

Школа - Инженерная школа информационных технологий и робототехники
 Направление подготовки – 01.03.02 Прикладная математика и информатика
 Отделение школы (НОЦ) – Отделение информационных технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка алгоритма решения задачи линейного программирования на основе операторов – проекторов и его программная реализация

УДК 004.021-047.84:519.852.3:517.98

Студент

Группа	ФИО	Подпись	Дата
8Б51	Бондаренко Павел Сергеевич		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Вылегжанин Олег Николаевич	Доцент, к.х.н.		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШБИП	Подопригора Игнат Валерьевич	Доцент, к.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ООТД	Мезенцева Ирина Леонидовна			

ДОПУСТИТЬ К ЗАЩИТЕ:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Руководитель ООП	Шевелев Геннадий Ефимович	Доцент, к.ф.-м.н.		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Код Результатов	Результат обучения (выпускник должен быть готов)	Требования ФГОС, Критерии АИОР
<i>Профессиональные компетенции</i>		
P1	Применять <i>глубокие математические и профессиональные знания</i> для решения задач научно-исследовательской, проектной, производственной и технологической деятельности в области системного и прикладного программирования.	<p>Требования ФГОС (ОК-11, 12, ПК-3, 10), Критерий 5 АИОР (п. 5.2.1), согласованный с требованиями международных стандартов <i>EUR-ACE</i> и <i>FEANI</i></p> <p>Требования профессиональных стандартов Ассоциации предприятий компьютерных и информационных технологий</p> <p>Требования работодателей: ФГУП «РФЯЦ-ВНИИТФ им. академика Е.И. Забабахина», ООО НАЦ «Недра», ИХН СО РАН</p>
P2	Умение использовать знания по естественнонаучным дисциплинам при определении задач математического моделирования объектов и явлений в различных предметных областях	<p>Требования ФГОС (ПК-3,9)</p> <p>Критерий 5 АИОР (п.5.2.3), согласованный с требованиями международных стандартов <i>EUR-ACE</i> и <i>FEANI</i></p> <p>Требования работодателей: ФГУП «РФЯЦ-ВНИИТФ им. академика Е.И. Забабахина», ООО «НАПО им. В.П. Чкалова», ИХН СО РАН</p>

P3	Демонстрировать понимание сущности и значения информации в развитии современного общества, владение основными методами, способами и средствами получения, хранения, переработки информации; использование для решения коммуникативных задач современных технических средств и информационных технологий.	Требования ФГОС (ОК-5, 11, 12,14,15, ПК-2, 6), Критерий 5 АИОР (п. 5.2.2), согласованный с требованиями международных стандартов <i>EUR-ACE</i> и <i>FEANI</i> Требования профессиональных стандартов Ассоциации предприятий компьютерных и информационных технологий Требования работодателей: Контек, ОАО «Газпром переработка», ООО Нижневартовскэнергонефть».
P4	Выполнять <i>инновационные</i> проекты с применением <i>глубоких профессиональных</i> знаний и <i>эффективных</i> методов проектирования для достижения <i>новых</i> результатов, обеспечивающих <i>конкурентные преимущества</i> в условиях экономических, экологических, социальных и других ограничений.	Требования ФГОС (ОК-14, ПК- 7, 9,14), Критерий 5 АИОР (п. 5.2.4), согласованный с требованиями международных стандартов <i>EUR-ACE</i> и <i>FEANI</i> Требования профессиональных стандартов Ассоциации предприятий компьютерных и информационных технологий. Требования работодателей: Контек, ОАО «Газпром переработка», ИХН СО РАН.
P5	<i>Демонстрировать знание</i> о формах организации образовательной и научной деятельности в высших учебных заведениях, <i>иметь навыки</i> преподавательской работы.	Требования ФГОС (ОК-1, 10, 16, ПК-1, 14, 15), Критерий 5 АИОР (п. 5.2.1), согласованный с требованиями международных стандартов <i>EUR-ACE</i> и <i>FEANI</i>
P6	Способность осуществлять организационно-управленческую и социально-ориентированную деятельность с соблюдением профессиональной этики	Требования ФГОС (ОК-5,13,16, ПК-11-13,16) Критерий 5 АИОР (п.5.2.12-13) согласованный с требованиями международных стандартов <i>EUR-ACE</i> и <i>FEANI</i>
Универсальные компетенции		

P7	<p><i>Активно владеть иностранным языком</i> на уровне, позволяющем работать в интернациональной среде, включая разработку документации и представление результатов инновационной деятельности. Толерантность в восприятии социальных и культурных различий.</p>	<p>Требования ФГОС (ОК-2, 3,4, 7, ПК-8).</p> <p>Критерий 5 АИОР (п. 5.2.11), согласованный с требованиями международных стандартов <i>EUR-ACE</i> и <i>FEANI</i> и Требования профессиональных стандартов Ассоциации предприятий компьютерных и информационных технологий</p>
P8	<p>Эффективно работать индивидуально, в качестве <i>члена и руководителя группы</i>, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность <i>следовать корпоративной культуре</i> организации</p>	<p>Требования ФГОС (ОК-1,4, 6, ПК-8,11,12),</p> <p>Критерий 5 АИОР (пп. 5.2.9,5.2.13), согласованный с требованиями международных стандартов <i>EUR-ACE</i> и <i>FEANI</i> и</p> <p>Требования профессиональных стандартов Ассоциации предприятий компьютерных и информационных технологий.</p> <p>Требования работодателей: Контек, ОАО «Газпром переработка», ООО Нижневартовскэнергогонефть».</p>
P9	<p><i>Самостоятельно учиться</i> и непрерывно <i>повышать квалификацию</i> в течение всего периода профессиональной деятельности.</p> <p>Способность к интеллектуальному, культурному, нравственному и профессиональному саморазвитию.</p>	<p>Требования ФГОС (ОК-8,9,16, ПК-5, 11),</p> <p>Критерий 5 АИОР (5.2.14), согласованный с требованиями международных стандартов <i>EUR-ACE</i> и <i>FEANI</i>.</p> <p>Требования работодателей: Контек, ОАО «Газпром переработка», ООО Нижневартовскэнергогонефть».</p>

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа - Инженерная школа информационных технологий и робототехники
Направление подготовки – 01.03.02 Прикладная математика и информатика
Уровень образования – Бакалавриат
Отделение школы (НОЦ) – Отделение информационных технологий
Период выполнения - осенний / весенний семестр 2018 /2019 учебного года

Форма представления работы:

Бакалаврская работа

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	
--	--

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (мо- дуля)
	Основная часть	75
	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
	Социальная ответственность	10

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Вылегжанин Олег Николаевич	Доцент, к.х.н.		

СОГЛАСОВАНО:

Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Руководитель ООП	Шевелев Геннадий Ефимович	Доцент, к.ф.- м.н.		



«Национальный исследовательский Томский политехнический университет» (ТПУ)

Отделение школы (НОЦ) – Отделение информационных технологий

(Подпись)
(Дата)
(Ф.И.О.)

на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8Б51	Бондаренко Павел Сергеевич

Тема работы:

Разработка алгоритма решения задачи линейного программирования на основе операторов – проекторов и его программная реализация.

Утверждена приказом директора (дата, номер)

Срок сдачи студентом выполненной работы:

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе

(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).

Предметом исследования являются задачи линейного программирования, такие как – задачи о смесях, транспортные задачи, сетевые задачи. Предложенный метод имеет ряд преимуществ перед обычно применяемым симплекс – методом.

<p>Перечень подлежащих исследованию, проектированию и разработке вопросов</p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<ol style="list-style-type: none"> 1. Изучить существующие способы решения задач линейного программирования; 2. Рассмотреть иные подходы к решению данных задач; 3. Провести эксперименты, по решению задач ЛП с использованием оператора - проектировщика; 4. Реализовать алгоритм решения задач ЛП; 5. Протестировать полученное программное обеспечение на примере конкретной задачи;
<p>Консультанты по разделам выпускной квалификационной работы</p> <p><i>(с указанием разделов)</i></p>	
<p>Раздел</p>	<p>Консультант</p>
<p>Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</p>	<p>Подопригора Игнат Валерьевич, доцент ОСГН ШБИП</p>
<p>Социальная ответственность</p>	<p>Мезенцева Ирина Леонидовна, ассистент ООТД</p>
<p>Названия разделов, которые должны быть написаны на русском и иностранном языках:</p>	
<p>Нет</p>	

<p>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</p>	
--	--

Задание выдал руководитель:

<p>Должность</p>	<p>ФИО</p>	<p>Ученая степень, звание</p>	<p>Подпись</p>	<p>Дата</p>
<p>Доцент ОИТ ИШИТР</p>	<p>Вылегжанин Олег Николаевич</p>	<p>Доцент, к.х.н.</p>		

Задание принял к исполнению студент:

<p>Группа</p>	<p>ФИО</p>	<p>Подпись</p>	<p>Дата</p>
<p>8Б51</p>	<p>Бондаренко Павел Сергеевич</p>		

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа - Инженерная школа информационных технологий и робототехники
 Направление подготовки – 01.03.02 Прикладная математика и информатика
 Уровень образования – Бакалавриат
 Отделение школы (НОЦ) – Отделение информационных технологий

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕ- РЕЖЕНИЕ»

Студенту:

Группа	ФИО
8Б51	Бондаренко Павел Сергеевич

Школа	ИШИТР	Отделение (НОЦ)	Информационных технологий
Уровень образования	Бакалавриат	Направление/специальность	Прикладная математика и информатика

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Человеческие ресурсы – 2 человека (руководитель и студент-дипломник).
--	---

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Планирование и формирование бюджета научных исследований	Необходимо составить календарный план НИ и рассчитать затраты НИ.
2. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Расчет интегральных показателей эффективности исследования, выбор наилучшего исполнения.

Перечень графического материала:

1. Оценка конкурентоспособности технических решений 2. Матрица SWOT 3. График проведения и бюджет НИ 4. Оценка ресурсной, финансовой и экономической эффективности НИ
--

Дата выдачи задания для раздела по линейному графику	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Подопригора Игнат Валерьевич	Кандидат экономических наук		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8Б51	Бондаренко Павел Сергеевич		

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа - Инженерная школа информационных технологий и робототехники
 Направление подготовки – 01.03.02 Прикладная математика и информатика
 Уровень образования – Бакалавриат
 Отделение школы (НОЦ) – Отделение информационных технологий

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8Б51	Бондаренко Павел Сергеевич

Школа	ИШИТР	Отделение (НОЦ)	Информационных технологий
Уровень образования	Бакалавриат	Направление/специальность	Прикладная математика и информатика

Исходные данные к разделу «Социальная ответственность»:

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения.	1. Алгоритм решения задач линейного программирования на основе оператора - проектора. Область применения: задачи о смесях, транспортные задачи.
---	---

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Производственная безопасность 1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности. 1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности.	1.1 Рассмотрены вредные факторы: Отклонение показателей микроклимата рабочей зоны; повышенный уровень шума на рабочем месте; отсутствие или недостаток естественного света, а также недостаточная освещенность рабочей зоны. 1.2 Рассмотрены опасные факторы: повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека; возгорание эксплуатируемого оборудования.
2. Экологическая безопасность: 2.1 Анализ воздействия объекта на окружающую среду; 2.2 Разработать решения по обеспечению экологической безопасности со ссылками на НТД по охране окружающей среды.	2.1 Рассмотрены негативно влияющие на экологию факторы при эксплуатации компьютера. 2.2 Решения по обеспечению экологической безопасности согласно нормативным документам.

3. Безопасность в чрезвычайных ситуациях: 3.1 Перечень возможных ЧС при разработке и эксплуатации проектируемого решения; 3.2 Разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.	3.1 Перечень возможных ЧС, которые могут возникнуть при работе в помещении офиса. 3.2 Способы защиты от пожара и ликвидация последствий.
4. Правовые и организационные вопросы обеспечения безопасности: 4.1 Организационные мероприятия при компоновке рабочей зоны.	4.1 Организационные мероприятия по обеспечению безопасности трудящихся за персональным компьютером.

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ООТД ШБИП	Мезенцева Ирина Леонидовна			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8Б51	Бондаренко Павел Сергеевич		

РЕФЕРАТ

Пояснительная записка содержит 75 страниц машинописного текста, 14 таблиц, 6 рисунков, 20 использованных источников, 10 приложений.

Целью работы является разработка и реализация алгоритма для решения задач линейного программирования на основе операторов – проекторов.

В данной работе был разработан и реализован алгоритм для решения задач линейного программирования, например, транспортных задач или задач о смесях. Программная реализация была осуществлена посредством высокоуровневого языка программирования C++.

Разработанный алгоритм позволяет быстрее и точнее решать задачи линейного программирования.

Ключевые слова: линейное программирование, оператор – проектор, псевдообратная матрица, обратная матрица, транспонированная матрица, градиент, целевая функция.

ОПРЕДЕЛЕНИЯ

В данной работе используются следующие термины с соответствующими определениями:

целевой функционал – функция, подлежащая оптимизации в рамках поставленной задачи;

оператор-проектор – матрица, построенная для заданного линейного пространства, результатом умножения произвольного вектора на которую является ортогональная пространству компонента этого вектора;

область допустимых значений – пространство решений задачи, образованное наложенными ограничениями;

гиперплоскость – пространство, имеющее размерность на единицу меньшую, чем исходное пространство;

псевдообратная матрица – обобщённая обратная матрица;

градиент функции – вектор, координатами которого являются частные производные функции;

рекуррентность – вычисляемость на основе предыдущих значений.

ОГЛАВЛЕНИЕ

ГЛАВА 1.ВВЕДЕНИЕ	15
ГЛАВА 2.ГЕОМЕТРИЧЕСКИЙ СМЫСЛ РЕШЕНИЯ ЗАДАЧИ.....	19
ГЛАВА 3.УЧЕТ ОГРАНИЧЕНИЙ РАВЕНСТВ И ПРЕОБРАЗОВАНИЕ ЦЕЛЕВОЙ ФУНКЦИИ С УЧЁТОМ ОГРАНИЧЕНИЙ.	23
ГЛАВА 4.КРИТЕРИЙ РЕШЕНИЯ	25
ГЛАВА 5.ПОИСК ОПТИМАЛЬНОГО РЕШЕНИЯ	27
ГЛАВА 6.ВЫВОДЫ.....	29
ГЛАВА 7. ПРАКТИЧЕСКАЯ ЧАСТЬ.....	30
ГЛАВА 8.СОЦИАЛЬНАЯ ОТВЕСТВЕННОСТЬ.....	32
8.1.Правовые и организационные вопросы обеспечения безопасности	32
8.2.Производственная безопасность.....	36
8.3.Анализ опасных и вредных производственных факторов	38
8.4.Обоснование мероприятий по снижению уровней воздействия опасных и вредных факторов на исследователя.	40
8.5.Экологическая безопасность.....	42
8.6.Безопасность в чрезвычайных ситуациях	42
8.7.Выводы и рекомендации.....	43
ГЛАВА 9.ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ	44
9.1 Оценка коммерческого и инновационного потенциала научных исследований.....	44
9.1.1 Потенциальные потребители результатов исследований	44
9.1.3 SWOT – анализ.....	46
9.3 Планирование научно-исследовательских работ.....	47

9.3.1 Структура работ в рамках научного исследования.....	47
9.4 Бюджет научно-технического исследования (НТИ).....	51
9.4.1 Расчет затрат на специальное оборудование для научных (экспериментальных) работ.....	51
9.4.2 Основная заработная плата исполнителей темы.....	51
9.4.3 Дополнительная заработная плата	52
9.4.4 Отчисления во внебюджетные фонды (страховые..... отчисления)	52
9.4.5 Накладные расходы.....	53
9.4.6 Формирование бюджета затрат научно-исследовательского проекта	53
9.5 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	54
10.СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	57
ПРИЛОЖЕНИЯ.....	59
Приложение А – Процедура перемножения матриц	59
Приложение Б – Вывод матриц на экран.....	60
Приложение В – Поиск столбца с максимальной нормой.....	61
Приложение Г – Копирование и удаление столбца.....	62
Приложение Д – Псевдоинверсия для столбца	63
Приложение Е – Расчёт псевдообратной матрицы.....	64
Приложение Ж – Расчёт проекции начальной точки на ОДЗ.....	65
Приложение З – Поиск базисного решения.....	66
Приложение И - Поиск точки оптимума	68
Приложение К – Тело программы.....	72

ГЛАВА 1. ВВЕДЕНИЕ

Линейное программирование является одной из основных частей того раздела современной математики, который получил название математического программирования. В общей постановке задачи этого раздела выглядят следующим образом [1].

Имеются какие-то переменные $x = (x_1, x_2, \dots, x_n)$ и функция этих переменных $f(x) = f(x_1, x_2, \dots, x_n)$, которая носит название целевой функции. Ставится задача: найти экстремум (максимум или минимум) целевой функции $f(x)$ при условии, что переменные x принадлежат некоторой области G :

$$\begin{cases} f(x) \rightarrow \text{extr} \\ x \in G \end{cases}$$

В зависимости от вида функции $f(x)$ и области G различают разделы математического программирования: квадратичное программирование, выпуклое программирование, целочисленное программирование и т.д.

Линейное программирование характеризуется тем, что

- а) Функция $f(x)$ является линейной функцией переменных x_1, x_2, \dots, x_n ;
- б) Область G определяется системой линейных равенств или неравенств.

Имеются некоторые стандартные формы задачи линейного программирования, к которым и приводят различные конкретные задачи.

Первая стандартная форма задачи линейного программирования имеет вид:

$$c_1x_1 + c_2x_2 + \dots + c_nx_n \rightarrow \max$$

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2$$

$$\dots\dots\dots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m$$

$$x_1 \geq 0; x_2 \geq 0; \dots x_n \geq 0;$$

Вторая стандартная форма задачи линейного программирования имеет вид:

$$\begin{aligned} c_1x_1 + c_2x_2 + \dots + c_nx_n &\rightarrow \min \\ a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &\geq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &\geq b_2 \\ &\dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &\geq b_m \\ x_1 \geq 0; x_2 \geq 0; \dots x_n &\geq 0; \end{aligned}$$

Канонической формой задачи линейного программирования называется задача вида:

$$\begin{aligned} c_1x_1 + c_2x_2 + \dots + c_nx_n &\rightarrow \min \\ a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ &\dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= b_m \\ x_1 \geq 0; x_2 \geq 0; \dots x_n \geq 0; \end{aligned}$$

Функцию $c_1x_1 + c_2x_2 + \dots + c_nx_n$ называют целевой функцией.

Любой набор чисел $\vec{x} = [x_1, x_2, \dots, x_n]^T$, удовлетворяющий ограничениям задачи, называют планом, а множество всех планов – допустимой областью. Тот план, который доставляет экстремум (минимум или максимум) целевой функции, называют оптимальным планом или просто решением задачи линейного программирования.

Правила приведения задач линейного программирования к стандартной и канонической формам.

Рассмотрим теперь те приёмы, которые позволяют произвольные формы задач линейного программирования приводить к указанным выше стандартным формам.

1. Превращение *max* в *min* и наоборот.

Если целевая функция в задаче линейного программирования задана в виде $c_1x_1 + c_2x_2 + \dots + c_nx_n \rightarrow \min$, то, умножая её на (-1) приведем к виду $-(c_1)x_1 + (-c_2)x_2 + \dots + (-c_n)x_n \rightarrow \max$, т.к. смена знака приводит к

2. Смена знака неравенства.

3. Превращение равенства в систему неравенств.

4. Превращение неравенств в равенства.

[illegible]

Здесь первые r ограничений имеют вид неравенств со знаком меньше либо равно $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i, i = \overline{1, r}$, затем идет группа неравенств со знаком больше либо равно $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \geq b_i, i = \overline{r+1, p}$ и, наконец, группа ограничений со знаком $=$. Для приведения задачи к канонической форме, где все ограничения имеют вид равенств, вводят дополнительные переменные $x_{n+1}, x_{n+2}, \dots, x_{n+p}$, которые тоже считаются неотрицательными и записывают исходную задачу в виде:

[illegible]

т.е. в неравенстве со знаком меньше либо равно добавляют дополнительную неотрицательную переменную, а из неравенства со знаком больше либо равно вычитают дополнительную переменную. В целевую функцию эти дополнительные переменные включают с коэффициентом 0, т.е. фактически они в целевой функции отсутствуют.

Для решения прикладных задач линейного программирования разработаны специальные методы линейного программирования, учитывающие регулярный характер задач данного типа и указанное свойство множества допустимых решений.

Возможные случаи допустимого множества решений задачи линейного программирования.

- 1) Допустимое множество решений пусто. Данному случаю соответствует взаимная противоречивость ограничений, входящих в задачу.
- 2) Допустимое множество - выпуклый ограниченный многогранник.
- 3) Допустимое множество - выпуклое неограниченное многогранное множество.

Два последних случая достаточно легко представить в двух- или трехмерном измерении. В пространстве большей размерности понятие многогранника (многогранного множества) вводится абстрактно как пересечение гиперплоскостей и гиперполуплоскостей, определяемых соответствующими линейными уравнениями и неравенствами, входящими в состав ограничений задачи.

Характерным свойством многогранника является наличие в нем особых точек - вершин.

Возможные случаи оптимальных решений (планов) задачи линейного программирования.

1) Задача не имеет оптимальных решений. Данный случай может возникнуть: либо тогда, когда допустимое множество решений пусто ("не из чего выбирать" оптимальный план), либо когда допустимое множество представляет собой неограниченное многогранное множество, и целевая функция на нем неограниченно возрастает (если $L \rightarrow \max$) или неограниченно убывает (при $L \rightarrow \min$).

2) Задача имеет единственное решение (единственный оптимальный план). Это решение обязательно совпадает с одной из вершин допустимого множества.

3) Задача имеет бесконечное множество оптимальных решений, заданное некоторым линейным образованием - ребром, гранью, гипергранью и т.д. Среди точек этого линейного образования имеются и вершины допустимого множества. Таким образом, основное утверждение теории линейного программирования, в конечном итоге определяющее специфические способы его решения, можно сформулировать следующим образом: Если задача линейного программирования имеет хотя бы один оптимальный план, то его следует искать среди вершин допустимого множества решений.

ГЛАВА 2. ГЕОМЕТРИЧЕСКИЙ СМЫСЛ РЕШЕНИЯ ЗАДАЧИ

В случае двух переменных модель задачи линейного программирования имеет следующий вид:

$$Z = c_1x_1 + c_2x_2 \rightarrow \max$$

$$a_1x_1 + a_2x_2 + b_i \geq 0,$$

$$x_1 \geq 0,$$

$$x_2 \geq 0.$$

(1)

Каждое ограничение представляет собой прямую, которая разбивает всё пространство (исходную плоскость) на две полуплоскости, одна из которых удовлетворяет ограничению (рис. 1.1).

Система ограничений представляет выпуклое множество, а в рассматриваемом двухмерном случае - выпуклый многоугольник ограничений (рис.1.2).

В частных случаях многоугольник может обращаться в точку (тогда решение единственно), прямую или отрезок. Если система ограничений противоречива (несовместна), то многоугольник ограничений построить нельзя и задача линейного программирования не имеет решений (рис.1.3).

Многоугольник ограничений, может быть, не замкнут (рис.1.4). В этом случае целевая функция не ограничена.

В случае n переменных каждое ограничение представляет $(n-1)$ -мерную гиперплоскость, которая делит все пространство на два полупространства. Система ограничений в этом случае дает выпуклый многогранник решений - общую часть n -мерного пространства, удовлетворяющую всем ограничениям.

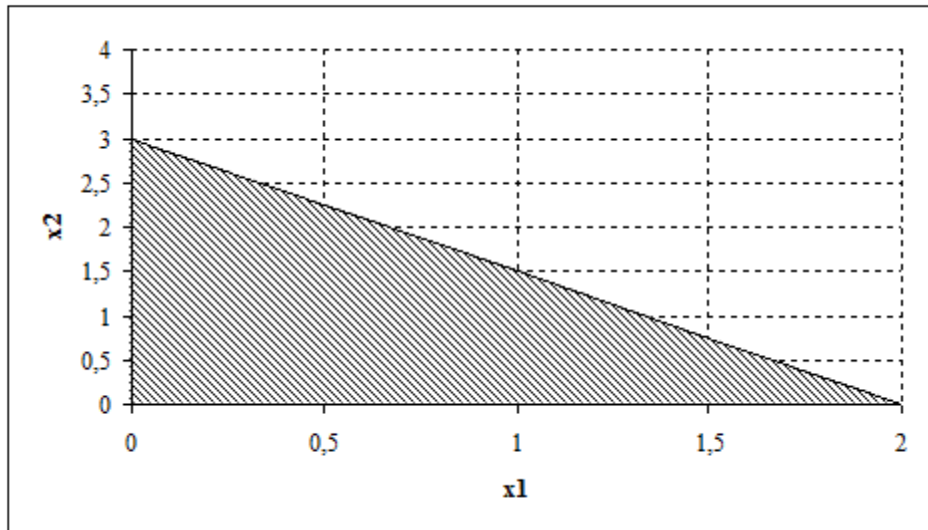


Рис.1.1. Геометрический смысл ограничения

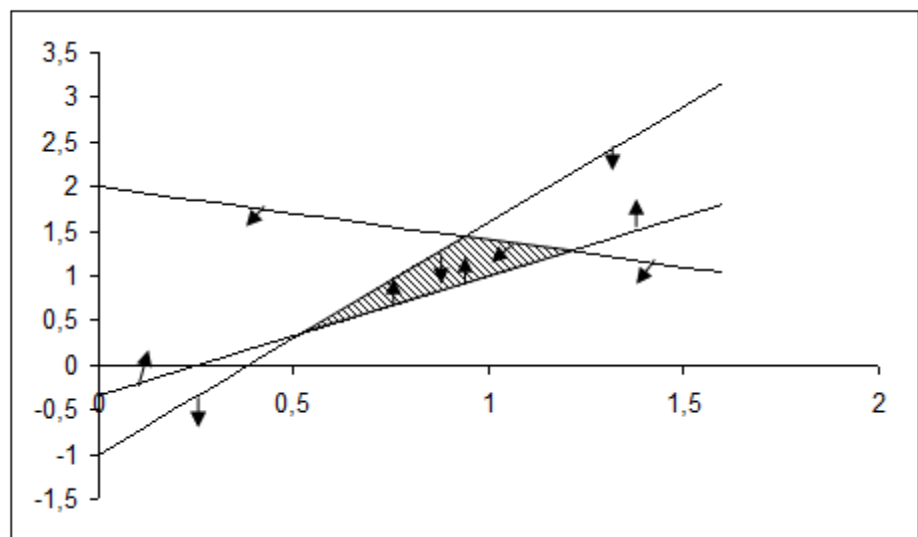


Рис.1.2. Геометрическая интерпретация системы ограничений

Для выяснения геометрического смысла целевой функции придадим переменной Z , различные числовые значения ($Z=0, Z=1, Z=2, \dots, Z=D$). Этим числовым значениям Z соответствует последовательность уравнений и система параллельных прямых в пространстве (рис.1.5).

$$c_1x_1 + c_2x_2 = 0,$$

$$c_1x_1 + c_2x_2 = 1,$$

...

$$c_1x_1 + c_2x_2 = D.$$

(2)

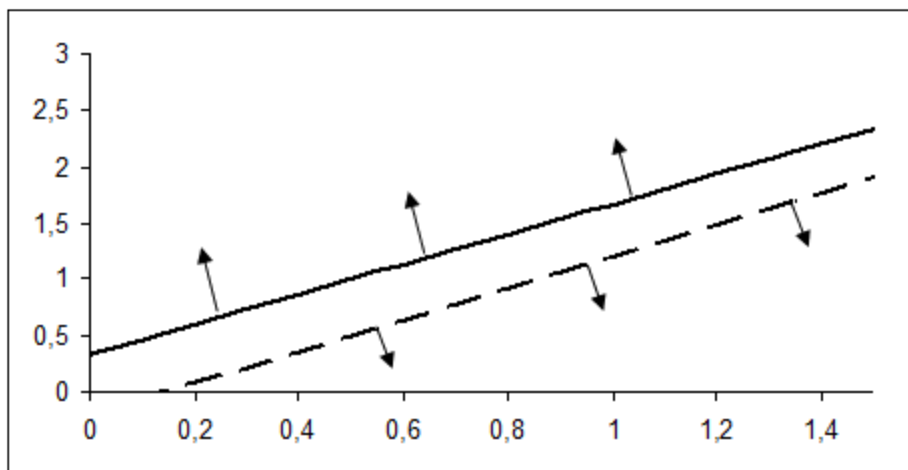


Рис.1.3. Несовместность системы ограничений

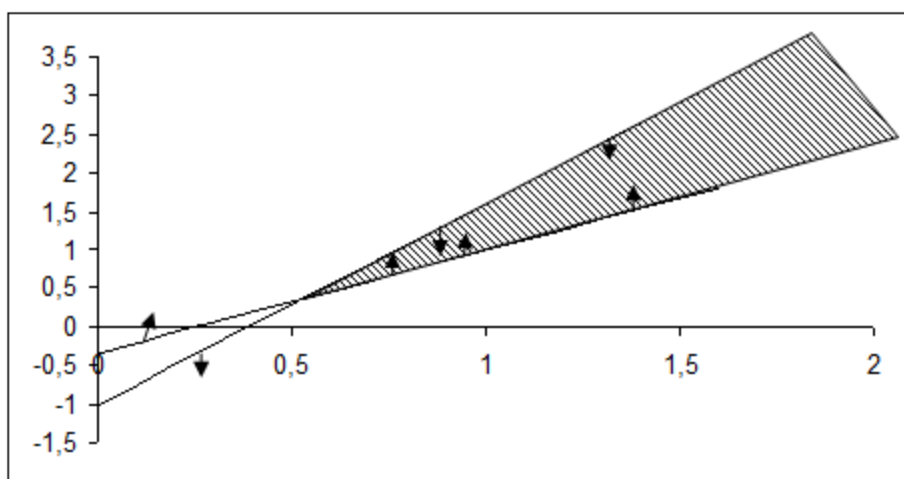


Рис.1.4. Неограниченность целевой функции

Первая прямая ($Z=0$) проходит через начало координат перпендикулярно (ортогонально) направляющему вектору $C = (c_1, c_2)$, последующие прямые параллельны первой и отстоят от нее в направлении вектора C на величину 1, 2, ..., D . В целом переменная Z определяет отклонение точек, лежащих на прямой $Z = c_1x_1 + c_2x_2$ от прямой $c_1x_1 + c_2x_2 = 0$, проходящей через начало координат. Для того чтобы определить отклонение любой точки от прямой $Z=0$, достаточно подставить координаты этой точки в уравнение целевой функции.

В n -мерном пространстве целевой функции, приравненной к нулю ($Z = c_1x_1 + c_2x_2 + \dots + c_jx_j + \dots + c_nx_n = 0$), геометрически соответствует $(n-1)$ -мерная гиперплоскость, проходящая через начало координат. Так как линейная форма Z достигает своего экстремального значения в крайней точке (вершине) многогранника ограничений, то геометрически задача линейного программирования заключается в отыскании вершины многогранника допустимых решений, имеющей максимальное отклонение от гиперплоскости, выраженной целевой функцией, приравненной к нулю (рис.1.6).

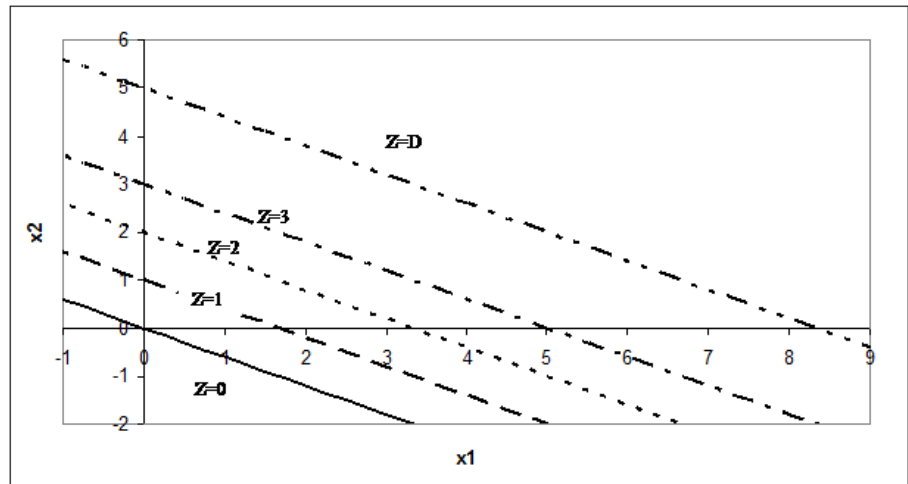


Рис.1.5.Геометрическая интерпретация целевой функции

Если многогранник ограничений не замкнут (рис.1.4), то уклонение равно бесконечности, так как прямую, параллельную целевой функции, можно перемещать сколько угодно вверх, не выходя за область допустимых значений переменных.

На геометрической интерпретации линейных задач основан графический метод их решения. Этот метод можно эффективно использовать при решении задач с двумя (иногда с тремя) переменными.

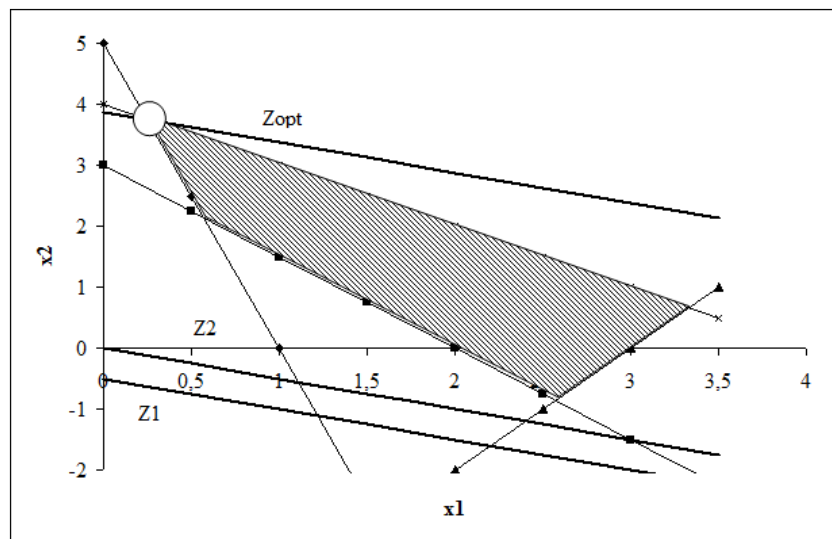


Рис.1.6. Геометрический смысл оптимального решения задачи линейного программирования

ГЛАВА 3.УЧЕТ ОГРАНИЧЕНИЙ РАВЕНСТВ И ПРЕОБРАЗОВАНИЕ ЦЕЛЕВОЙ ФУНКЦИИ С УЧЁТОМ ОГРАНИЧЕНИЙ.

При решении задач оптимизации с ограничениями ограничения являются линейными равенствами или неравенствами. Учет ограничений определяет

метод решения задачи, её «трудоемкость». Учет ограничений приводит к сокращению размерности пространства. Алгоритм, который был выбран для учета ограничений – равенств, основан на процедуре рекуррентного псевдообращения.

Постановка задачи оптимизации при наличии ограничений – равенств может быть сформулирована следующим образом:

$$f(x) = \langle c, x \rangle \rightarrow \text{extr},$$

$$A \cdot x = b, x \in \mathbb{R}^n, b \in \mathbb{R}^m, \text{rk} A = k.$$

Будем полагать, что первые k – столбцов A – линейно независимы. Тогда одним из подходов к решению является отыскание такого оператора P , при умножении на который матрица PA имеет вид:

$$P \cdot A = [D \cdot A'],$$

Где D – диагональная матрица размером $k \times k$, A' – некоторая матрица размером $k \times (n-k)$.

Цель этого преобразования – выразить первые k неизвестных как линейные функции от остальных.

Введем следующие обозначения: \hat{x} – вектор первых k элементов, \bar{x} – вектор остальных элементов x ; \hat{A} – матрица, образованная первыми k столбцами из A , \bar{A} – матрица, образованная остальными $(n-k)$ столбцами. Тогда получить требуемое преобразование можно на основе следующего утверждения.

Утверждение 1. Если A – матрица ограничений - равенств, $\text{rk} A = k$ и первые k столбцов матрицы A линейно независимы, то систему ограничений можно записать в виде:

$$\hat{x} = \hat{A}^+(b - \bar{A} * \bar{x})$$

Где \hat{A}^+ – матрица, псевдообратная к матрице A .

Доказательство 1.

С учетом введенных обозначений систему ограничений можно представить в виде: $\hat{A} \times \hat{x} + \bar{A} \times \bar{x} = b$.

Умножив обе части на \hat{A}^+ получим:

$$A^+ \cdot A \cdot x + A^+ \cdot \bar{A} \cdot \bar{x} = A^+ \cdot b \quad \text{или} \quad I \cdot x = A^+ \cdot b - A^+ \cdot \bar{A} \cdot \bar{x}$$

Где I – единичная матрица, что и требовалось получить.

При этом целевая функция преобразуется в соответствии с следующим утверждением.

Утверждение 2. Если $\langle c, x \rangle$ - линейная форма, то при наличии ограничительных равенств вида $A_1 x = b_1$ она преобразуется к форме

$$c^T \cdot x = \hat{c}^T \cdot \hat{A}^+ \cdot b + (\bar{c}^T - \hat{c}^T \cdot \hat{A}^+ \cdot \bar{A}) \cdot \bar{x}$$

Где \hat{c} и \bar{c} – компоненты вектора коэффициентов, соответствующие компонентам \hat{x} и \bar{x} вектора x .

Доказательство 2.

Разлагая компоненты линейной формы на составляющие, имеем:

$$c^T \cdot x = \hat{c}^T \cdot \hat{x} + \bar{c}^T \cdot \bar{x}$$

Заменив $\hat{x} = \hat{A}^+ (b - \bar{A} \cdot \bar{x})$, получим

$$\hat{c}^T \cdot A^+ (b - \bar{A} \cdot \bar{x}) + \bar{c}^T \cdot \bar{x} = \hat{c}^T \cdot A^+ b - \hat{c}^T \cdot A^+ \bar{A} \cdot \bar{x} + \bar{c}^T \cdot \bar{x},$$

Откуда, вынося в правой части за скобки общий множитель \bar{x} – получим доказываемое.

ГЛАВА 4. КРИТЕРИЙ РЕШЕНИЯ

Для обоснования алгоритма получения опорного решения докажем две теоремы.

Утверждение 3. Если x_0 – начальная точка, а ближайшая к ней плоскость многогранника M имеет направляющим вектором a_i , то x_1 – проекция начальной точки на эту плоскость есть:

$$x_1 = x_0 + \frac{b_i - a_i^T \cdot x_0}{a_i^T \cdot a_i} \cdot a_i, \quad (3)$$

Доказательство 3. Преобразуем (3) к виду:

$$x_1 - x_0 = \frac{b_i - a_i^T \cdot x_0}{a_i^T \cdot a_i} \cdot a_i. \quad (4)$$

Так как $\frac{\mathbf{b}_i - \mathbf{a}_i^T \cdot \mathbf{x}_0}{\mathbf{a}_i^T \cdot \mathbf{a}_i}$ – скаляр, то направление из \mathbf{x}_0 на точку \mathbf{x}_1 совпадает

с направляющим вектором плоскости \mathbf{a}_i , т. е. нормально к этой плоскости.

Умножим обе части (3) слева на \mathbf{a}_i^T получим:

$$\mathbf{a}_i^T \cdot \mathbf{x}_1 = \mathbf{a}_i^T \cdot \mathbf{x}_0 + \frac{\mathbf{b}_i - \mathbf{a}_i^T \cdot \mathbf{x}_0}{\mathbf{a}_i^T \cdot \mathbf{a}_i} \cdot \mathbf{a}_i^T \cdot \mathbf{a}_i = \mathbf{b}_i, \quad (5)$$

т. е. \mathbf{x}_1 принадлежит плоскости, заданной направляющим вектором \mathbf{a}_i .

Утверждение 4. Если точка \mathbf{x}_k принадлежит пересечению k плоскостей, заданных направляющими векторами $\mathbf{a}_j | j=1, \dots, k$, то ее проекция на пересечение плоскости, заданной строкой \mathbf{a}_i , с пересечением плоскостей, заданных строками $\mathbf{a}_j | j=1, \dots, k$, составляющими матрицу \mathbf{A}_k , есть:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{\mathbf{b}_i - \mathbf{a}_i^T \cdot \mathbf{x}_k}{\mathbf{a}_i^T \cdot \mathbf{R} \cdot \mathbf{a}_i} \cdot \mathbf{R} \cdot \mathbf{a}_i, \quad (6)$$

где $\mathbf{R} = (\mathbf{I} - \mathbf{A}_k \cdot \mathbf{A}_k^+)$ – оператор-проектор на пространство, перпендикулярное пространству, натянутому на строки матрицы \mathbf{A}_k .

Доказательство 4.

Преобразуем (6) к виду:

$$\mathbf{x}_{k+1} - \mathbf{x}_k = \frac{\mathbf{b}_i - \mathbf{a}_i^T \cdot \mathbf{x}_k}{\mathbf{a}_i^T \cdot \mathbf{R} \cdot \mathbf{a}_i} \cdot \mathbf{R} \cdot \mathbf{a}_i. \quad (7)$$

Так как $\frac{\mathbf{b}_i - \mathbf{a}_i^T \cdot \mathbf{x}_k}{\mathbf{a}_i^T \cdot \mathbf{R} \cdot \mathbf{a}_i}$ – скаляр, то направление из точки \mathbf{x}_k на \mathbf{x}_{k+1} перпендикулярно подпространству, натянутому на строки матрицы \mathbf{A}_k .

1. Умножим обе части (6) слева на \mathbf{a}_i^T :

$$\mathbf{a}_i^T \cdot \mathbf{x}_{k+1} = \mathbf{a}_i^T \cdot \mathbf{x}_k + \frac{\mathbf{b}_i - \mathbf{a}_i^T \cdot \mathbf{x}_k}{\mathbf{a}_i^T \cdot \mathbf{R} \cdot \mathbf{a}_i} \cdot \mathbf{a}_i^T \cdot \mathbf{R} \cdot \mathbf{a}_i = \mathbf{b}_i. \quad (8)$$

Следовательно, точка x_{k+1} принадлежит плоскости с направляющим вектором a_i .

2. Умножая (6) на матрицу A_k слева, получим:

$$A_k \cdot x_{k+1} = A_k \cdot x_k + \frac{b_i - a_i^T \cdot x_k}{a_i^T \cdot R \cdot a_i} \cdot A_k \cdot R \cdot a_i = A_k \cdot x_k, \quad (9)$$

так как $R \cdot a_i$ по определению принадлежит подпространству, перпендикулярному пересечению плоскостей, задаваемых строками матрицы A_k , то $A_k \cdot R \cdot a_i = 0$.

Утверждения 3 и 4 обосновывают следующий процесс получения опорного решения задачи, т. е. получения точки, принадлежащей вершине многогранника M . Выбрав произвольную начальную точку x_0 , перейдем к следующей точке на плоскости, ближайшей к x_0 , а затем выполним последовательно (4) для тех i , для которых $y_i = b_i - a_i^T \cdot x_k \leq 0$, а $\frac{b_i - a_i^T \cdot x_k}{a_i^T \cdot a_i}$ – минимален. В результате получим точку, для которой $\forall y_i \leq 0 \quad i = 1 \dots n$, т. е. являющуюся вершиной многогранника M .

ГЛАВА 5. ПОИСК ОПТИМАЛЬНОГО РЕШЕНИЯ

Поиск оптимума целевой функции осуществляется путем перемещения из одной вершины многогранника M в другую в направлении градиента целевой функции. Для такого перехода необходимо в точке опорного решения выбрать ребро многогранника ограничений, которое имеет наибольшую проекцию на градиент целевой функции.

Пусть A_k - матрица ограничений, активных в очередной точке (вершине) M . Тогда ребро многогранника M , имеющее наибольшую проекцию на градиент целевой функции, образованное пересечением $k-1$ плоскости, приводит в новую вершину многогранника, которая находится ближе к оптимуму целевой функции.

Вычислим в качестве вспомогательного вектор:

$$\lambda = (A^T)^+ \cdot c. \quad (10)$$

Элементы вектора λ есть коэффициенты функции Лагранжа, т.е. величины приращения целевой функции при перемещении вдоль соответствующего направления. Поэтому при выборе исключаемой строки матрицы A_k надо брать ту, для которой эта компонента будет максимальна по модулю среди отрицательных.

Получение оптимального решения связано с перемещением рабочей точки из этой вершины в ту, для которой значение целевой функции оптимально. Для шага перемещения необходимо из набора k ограничений-неравенств (строк матрицы A_k), которые обращаются в равенства в этой вершине, исключить одно неравенство и добавить другое. Соответственно, необходимо пересчитать новую матрицу A_k^+ . Для пересчета можно использовать процедуру рекуррентного псевдообращения [5]. Обозначим C_k матрицу, содержащую k столбцов, а c_k – добавляемый столбец.

Тогда, согласно [8]:

$$C_k = (C_{k-1}, c_k), \quad (11)$$

$$C_k^+ = \begin{pmatrix} C_{k-1}^+ [I - c_k \cdot c_k^+] \\ h_k \end{pmatrix}, \quad (12)$$

$$\text{где } h_k = \frac{c_k^T \cdot (I - C_{k-1} \cdot C_{k-1}^+)}{c_k^T \cdot (I - C_{k-1} \cdot C_{k-1}^+) \cdot c_k}.$$

Далее, в соответствии с [8]:

$$C_{k-1}^+ = H_{k-1} \cdot (I - d_k \cdot h_k), \quad (13)$$

$$\text{где } d_k = h_k^T / h_k \cdot h_k^T.$$

Вычислим коэффициенты Лагранжа $\lambda = (A_k^T)^+ \cdot c$. Обозначим J – множество индексов соответствующих неравенствам, включенным в A_k^+ . Для выбора неравенства a_i , исключаемого из J на первом шаге, нужно взять максимальное среди положительных значений коэффициентов Лагранжа.

Номер неравенства, включение которого в J соответствует новой вершине, есть номер минимального положительного значения коэффициентов Лагранжа:

$$\alpha = \frac{\mathbf{b}_i - \mathbf{a}_i^T \cdot \mathbf{x}_k}{\mathbf{a}_i^T \cdot \Omega \cdot \mathbf{a}_i}. \quad (14)$$

Решению задачи соответствует точка \mathbf{x}^* , для которой выполняются все условия Куна – Таккера:

1. $\mathbf{x}^* \in M$, где \mathbf{x}^* – решение задачи,
2. $\lambda^i * (b_i - \mathbf{a}_i^T \cdot \mathbf{x}_i) = 0, i = 1, \dots, n$
3. $\mathbf{c} + A^T \lambda = 0$

Где λ_i – вектор неизвестных, неотрицательных коэффициентов (коэффициентов Лагранжа), \mathbf{a}_i – строки матрицы ограничений – неравенств, а M – область выполнения ограничений – неравенств (область допустимых решений).

Причем первое и третье (градиент функции Лагранжа) условия выполняются по построению.

ГЛАВА 6. ВЫВОДЫ

Данный алгоритм имеет ряд преимуществ перед обычно применяемым симплекс – методом Данцига:

1. За счет анализа ограничений равенств, существенно сокращается размерность пространства решения задачи.
2. В обычно применяемом методе Данцига каждое ограничение - равенство вида $\mathbf{a}^T \mathbf{x} = b$ преобразуется в два ограничения неравенства ($\mathbf{a}^T \mathbf{x} \leq b, \mathbf{a}^T \mathbf{x} \geq b$), что приводит к увеличению числа вершин, которое равно числу сочетаний из n по k , где n – число ограничений неравенств, а k - размерность пространства решения задачи. Увеличение числа вершин многогранника допустимых значений, в свою очередь, может привести к увеличению числа шагов перемещения рабочей точки в точку оптимума.
3. Увеличение числа шагов в поиске оптимума, в свою очередь, приводит к увеличению ошибки округления решения задачи

ГЛАВА 7. ПРАКТИЧЕСКАЯ ЧАСТЬ

Для проверки работоспособности реализованного алгоритма рассмотрим, в качестве примера, задачу со следующими входными данными:

$$\begin{array}{l}
 A1 = \begin{pmatrix} -0.91 & -0.868 & 1.097 & -1.023 & -1.252 \\ 0.625 & -0.738 & 0.541 & 0.664 & 0.52 \\ 0.146 & -0.051 & 0.316 & 0.249 & -0.171 \\ -0.453 & 0.294 & 0.151 & -0.934 & 0.793 \end{pmatrix} \quad B1 = \begin{pmatrix} 0.716 & 0.386 \\ 1.044 & -1.429 \\ 0.811 & -0.411 \\ 0.617 & 0.874 \\ & -0.569 \end{pmatrix} \\
 A2 = \begin{pmatrix} -0.245 & -0.086 & 0.685 & 0.791 & 0.358 \\ -0.211 & -0.156 & 0.136 & 0.864 & 1.958 \\ 0.069 & 0.565 & 0.557 & 1.891 & 1.479 \\ -2.294 & 0.662 & -0.441 & -0.474 & 0.089 \\ -0.699 & 0.347 & -0.875 & 1.9 & -2.726 \\ 0.87 & 2.179 & 1.085 & 1.176 & -0.971 \\ 0.496 & -0.802 & -0.627 & -0.553 & -0.346 \\ -0.464 & 0.251 & -0.373 & -0.273 & 1.29 \end{pmatrix} \quad B2 = \begin{pmatrix} 0.194 \\ 2.086 \\ -1.562 \\ 0.079 \\ 0.42 \\ 1.627 \\ 0.62 \\ -1.717 \end{pmatrix} \\
 X0 = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \\ 0.5 \end{pmatrix}
 \end{array}$$

При учете ограничений равенств матрицы ограничений и целевого функционала примут следующий вид:

$$\begin{array}{l}
 A1^* = \begin{pmatrix} -0.91 & -0.868 & 1.097 & 0 & 0 \\ 0.625 & -0.738 & 0.541 & 0 & 0 \\ 0.146 & -0.051 & 0.316 & 0 & 0 \\ -0.453 & 0.294 & 0.151 & 0 & 0 \end{pmatrix} \quad A1^*n = \begin{pmatrix} -1.252 & -1.023 \\ 0.52 & 0.664 \\ -0.171 & 0.249 \\ 0.793 & -0.934 \end{pmatrix} \\
 A2^* = \begin{pmatrix} -0.245 & -0.086 & 0.685 & 0 & 0 \\ -0.211 & -0.156 & 0.136 & 0 & 0 \\ 0.069 & 0.565 & 0.557 & 0 & 0 \\ -2.294 & 0.662 & -0.441 & 0 & 0 \\ -0.699 & 0.347 & -0.875 & 0 & 0 \\ 0.87 & 2.179 & 1.085 & 0 & 0 \\ 0.496 & -0.802 & -0.627 & 0 & 0 \\ -0.464 & 0.251 & -0.373 & 0 & 0 \end{pmatrix} \quad A2^*n = \begin{pmatrix} 0.358 & 0.791 \\ 1.958 & 0.864 \\ 1.479 & 1.891 \\ 0.089 & -0.474 \\ -2.726 & 1.9 \\ -0.971 & 1.176 \\ -0.346 & -0.553 \\ 1.29 & -0.273 \end{pmatrix}
 \end{array}$$

$$Pn = 0.874$$

Можно сделать вывод о том, что размерность пространства сократилась с 5 до 3.

Теперь, изобразив новые ограничения – неравенства, получим много-

гранник допустимых решений.

Далее необходимо найти базисное решение. Для этого введём начальную точку:

$$\begin{array}{l} 0.1 \\ X_0 = 0.3 \\ 0.5 \end{array}$$

Тогда следующей точкой будет проекция на ближайшую поверхность многогранника допустимых решений:

$$\begin{array}{l} 0.367256 \\ X_1 = -2.85434 \\ 2.22561 \end{array}$$

Теперь необходимо найти ближайшую вершину, от которой будет произведен поиск точки оптимума:

$$\begin{array}{l} 1.94522 \\ X_b = -0.836803 \\ 1.62033 \end{array}$$

Для нахождения оптимального решения нужно найти такую точку, которая будет удовлетворять всем условиям Куна – Таккера. Таковой будет являться точка с координатами:

$$\begin{array}{l} 1.47995 \\ \text{Optimal } X = -2.59469 \\ 1.11473 \end{array}$$

Для проверки подставим в целевой функционал координаты данной точки. В результате должны получить 2.7.

$$I(x) = p^T x = 2.68991$$

Таким образом решение найдено верно, что говорит о готовности программы к решению задач линейного программирования.

ГЛАВА 8. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

Выполненная работа направлена на построение алгоритма, который будет решать задачи линейного программирования с использованием оператора – проектора.

Данная работа имеет высокий потенциал. Описанный в работе алгоритм может использоваться для, например, решения транспортных задач, задач о смесях и других задач линейного программирования. Данный способ имеет ряд преимуществ перед обычно применяемым симплекс – методом Дж.Данцига, такие как:

1. За счет анализа ограничений равенств, существенно сокращается размерность пространства решения задачи.

2. В обычно применяемом методе Данцига каждое ограничение - равенство вида $aTx=b$ преобразуется в два ограничения неравенства ($aTx \leq b$, $aTx \geq b$), что приводит к увеличению числа вершин, которое равно числу сочетаний из n по k , где n – число ограничений неравенств, а k - размерность пространства решения задачи. Увеличение числа вершин многогранника допустимых значений, в свою очередь, может привести к увеличению числа шагов перемещения рабочей точки в точку оптимума.

3. Увеличение числа шагов в поиске оптимума, в свою очередь, приводит к увеличению ошибки округления решения задачи

Все действия, описанные в данной выпускной квалификационной работе, были совершены в городе Томск, на территории кампуса томского политехнического университета.

8.1. Правовые и организационные вопросы обеспечения безопасности

Сотруднику, постоянно работающему за компьютером, очень нелегко долгое время концентрировать взгляд на экране монитора. Устают глаза, шея, спина, затекают ноги. Для предотвращения негативных последствий воздействия на организм необходимо делать перерывы в работе. Законодательно установление перерывов закреплено в статье 109 Трудового Кодекса РФ. Правда,

указано, что виды работ, предусматривающих предоставление работникам в течение рабочего времени специальных перерывов, а также их продолжительность устанавливаются правилами внутреннего трудового распорядка. Конкретно про перерывы при работе за компьютером ничего не сказано.

Однако, по типовой инструкции по охране труда при работе на персональном компьютере ТОО Р-45-084-01 всё же вменяется установление перерывов. В целом продолжительность непрерывной работы за компьютером не должна превышать 2-х часов.

Следует обратить внимание на то, что основная работа за компьютером предусматривает не менее 50 % времени в течение рабочей смены или рабочего дня нахождения за ним. Время перерыва зависит от вида и сложности осуществляемой работы путем деления на группы. Выделяют 3 группы: А (работа по считыванию информации с экрана компьютера с предварительным запросом), Б (работа по вводу информации), В (творческая работа в режиме диалога с компьютером).

В зависимости от сложности работы, установление числа и длительности перерывов происходит следующим образом:

- 1) для группы А (не свыше 60000 считываемых знаков за смену) перерыв составляет 15 минут, предоставляется два раза – через два часа после начала работы и перерыва на обед;
- 2) для группы Б (не свыше 40000 вводимых знаков за смену) перерыв составляет 10 минут через каждый трудовой час;
- 3) для группы В (не свыше шести 6 часов за смену) перерыв составляет 15 минут через каждый трудовой час.

В целях обеспечения прав и свобод человека и гражданина работодатель и его представители при обработке персональных данных работника обязаны соблюдать следующие общие требования:

- 1) обработка персональных данных работника может осуществляться исключительно в целях обеспечения соблюдения законов и иных нормативных правовых актов, содействия работникам в трудоустройстве, получении образо-

вания и продвижении по службе, обеспечения личной безопасности работников, контроля количества и качества выполняемой работы и обеспечения сохранности имущества;

2) при определении объема и содержания обрабатываемых персональных данных работника работодатель должен руководствоваться Конституцией Российской Федерации, настоящим Кодексом и иными федеральными законами;

3) все персональные данные работника следует получать у него самого. Если персональные данные работника возможно получить только у третьей стороны, то работник должен быть уведомлен об этом заранее и от него должно быть получено письменное согласие. Работодатель должен сообщить работнику о целях, предполагаемых источниках и способах получения персональных данных, а также о характере подлежащих получению персональных данных и последствиях отказа работника дать письменное согласие на их получение;

4) работодатель не имеет права получать и обрабатывать сведения о работнике, относящиеся в соответствии с законодательством Российской Федерации в области персональных данных к специальным категориям персональных данных, за исключением случаев, предусмотренных настоящим Кодексом и другими федеральными законами;

5) работодатель не имеет права получать и обрабатывать персональные данные работника о его членстве в общественных объединениях или его профсоюзной деятельности, за исключением случаев, предусмотренных настоящим Кодексом или иными федеральными законами;

6) при принятии решений, затрагивающих интересы работника, работодатель не имеет права основываться на персональных данных работника, полученных исключительно в результате их автоматизированной обработки или электронного получения;

7) защита персональных данных работника от неправомерного их использования или утраты должна быть обеспечена работодателем за счет его средств в порядке, установленном настоящим Кодексом и иными федеральными

ми законами;

8) работники и их представители должны быть ознакомлены под роспись с документами работодателя, устанавливающими порядок обработки персональных данных работников, а также об их правах и обязанностях в этой области;

9) работники не должны отказываться от своих прав на сохранение и защиту тайны;

10) работодатели, работники и их представители должны совместно выработать меры защиты персональных данных работников.

Согласно ТК РФ Статья 135 об установлении заработной платы, заработная плата работнику устанавливается трудовым договором в соответствии с действующими у данного работодателя системами оплаты труда.

Системы оплаты труда, включая размеры тарифных ставок, окладов (должностных окладов), доплат и надбавок компенсационного характера, в том числе за работу в условиях, отклоняющихся от нормальных, системы доплат и надбавок стимулирующего характера и системы премирования, устанавливаются коллективными договорами, соглашениями, локальными нормативными актами в соответствии с трудовым законодательством и иными нормативными правовыми актами, содержащими нормы трудового права.

Локальные нормативные акты, устанавливающие системы оплаты труда, принимаются работодателем с учетом мнения представительного органа работников.

Условия оплаты труда, определенные трудовым договором, не могут быть ухудшены по сравнению с установленными трудовым законодательством и иными нормативными правовыми актами, содержащими нормы трудового права, коллективным договором, соглашениями, локальными нормативными актами. Условия оплаты труда, определенные коллективным договором, соглашениями, локальными нормативными актами, не могут быть ухудшены по сравнению с установленными трудовым законодательством и иными нормативными правовыми актами, содержащими нормы трудового права.

Что же касается нормы выработки, времени, нормативы численности и других норм – они устанавливаются в соответствии с достигнутым уровнем техники, технологии, организации производства и труда. Нормы труда могут быть пересмотрены по мере совершенствования или внедрения новой техники, технологии и проведения организационных либо иных мероприятий, обеспечивающих рост производительности труда, а также в случае использования физически и морально устаревшего оборудования. Достижение высокого уровня выработки продукции (оказания услуг) отдельными работниками за счет применения по их инициативе новых приемов труда и совершенствования рабочих мест не является основанием для пересмотра ранее установленных норм труда.

Конкретные размеры повышения оплаты труда устанавливаются работодателем с учетом мнения представительного органа работников в порядке, установленном статьей 372 настоящего Кодекса для принятия локальных нормативных актов, либо коллективным договором, трудовым договором.

Работодателю необходимо предоставить обязательное социальное страхование работников от несчастных случаев на производстве и профессиональных заболеваний. ИП и организации, являющиеся работодателями, обязаны с выплат сотрудникам, работающим по трудовым договорам, ежемесячно перечислять страховые взносы на пенсионное, медицинское и социальное страхование в ФНС РФ.

Под проектированием рабочего места понимается целесообразное пространственное размещение в горизонтальной и вертикальной плоскостях функционально взаимосвязанных средств производства (оборудования, оснастки, предметов труда и др.), необходимых для осуществления трудового процесса. При проектировании рабочих мест должны быть учтены освещенность, температура, влажность, давление, шум, наличие вредных веществ, электромагнитных полей и другие санитарно-гигиенические требования к организации рабочих мест.

8.2.Производственная безопасность

Опасным производственным фактором (ОПФ) называется такой произ-

водственный фактор, воздействие которого на работающего мгновенно приводит к травме или летальному исходу. Травма – это повреждение тканей организма и нарушение его функций внешним воздействием.

Вредным производственным фактором (ВПФ) называется такой производственный фактор, воздействие которого на работающего в определенных условиях приводит к заболеванию или снижению трудоспособности.

Таблица 1 - Возможные опасные и вредные факторы

Факторы (ГОСТ 12.0.003-2015)	Этапы работ		Нормативные документы
	Разработка	Эксплуатация	
1.Отклонение показателей микроклимата	+	+	СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений
2. Превышение уровня шума		+	СН 2.2.4/2.1.8.562-96. Санитарные нормы. Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки.
3.Отсутствие или недостаток естественного света	+	+	СанПиН 2.2.1/2.1.1.1278-03 Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий.
4.Недостаточная освещенность рабочей зоны		+	СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95*[59].

5.Повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека	+	+	СанПиН 2.2.2/2.4.1340-03 "Гигиенические требования к персональным электронно-вычислительным машинам и организации работы" (с изменениями на 21 июня 2016 года)
--	---	---	--

8.3. Анализ опасных и вредных производственных факторов

На отклонение показателей микроклимата влияют влажность, запыленность, температура воздуха. Если температура выше нормы, кровеносные сосуды расширяются и теплоотдача в окружающую среду возрастает, если ниже, то сосуды сужаются, приток крови к телу замедляется и теплоотдача уменьшается. Слишком высокая влажность (более 85 %) затрудняет терморегуляцию, а слишком низкая (менее 20 %) вызывает пересыхание слизистых, причем не только дыхательных путей, но и глаз. Чем выше влажность в помещении, тем слабее влияние электростатических и электромагнитных полей, уровень излучения которых в помещении, где установлен компьютер, всегда повышен. Чтобы узнать необходимые параметры микроклимата помещения, нужно знать категорию интенсивности энергозатрат организма. Исполнители данной работы относятся к Ia-категории с интенсивностью энергозатрат до 120 ккал/ч (до 139 Вт), так как представленный вид работ выполняется сидя и сопровождается незначительным физическим напряжением. Выполненная работа проводилась в теплый период года. Исходя из заданных параметров, температура воздуха должна составлять 22-24 °С, температура поверхностей: 21-25 °С, относительная влажность воздуха – 60-40%, а скорость движения воздуха – 0.1 м/с. Допустимые нормы: перепад температуры воздуха по высоте должен быть не более 3° С; перепад температуры воздуха по горизонтали, а также ее изменения в течение смены не должны превышать 4° С; при температуре воздуха на рабочих местах 25° С и выше максимально допустимые величины относительной влажности воздуха не должны выходить за пределы 70% - при температуре воздуха 25°С. Допустимые величины интенсивности теплового облучения работающих от источников излучения, нагретых до белого и красного свечения (раскален-

ный или расплавленный металл, стекло, пламя и др.) не должны превышать 140 Вт/кв.м. При этом облучению не должно подвергаться более 25% поверхности тела и обязательным является использование средств индивидуальной защиты, в том числе средств защиты лица и глаз.

1) Источниками шума являются сами вычислительные машины, центральная система вентиляции и кондиционирования воздуха и другое оборудование. Работающие в условиях длительного шумового воздействия люди испытывают раздражительность, головокружение, снижение памяти, повышенную утомляемость. Под воздействием шума снижается концентрация внимания, нарушаются физиологические функции, ухудшается речевая коммутация. Все это снижает работоспособность человека и его производительность, качество и безопасность труда. При выполнении основной работы уровень шума на рабочем месте не должен превышать 50 дБА. Дополнительными мероприятиями по шумогашению в машинных залах могут быть устройство подвесного потолка, который служит звукопоглощающим экраном, использование звукопоглощающих материалов с максимальными коэффициентами звукопоглощения в области частот 63-8000 Гц для отделки помещений, уменьшение площади стеклянных ограждений и оконных проемов, так же можно использовать звукогасящие экраны на рабочем месте и однотонные занавески из плотной ткани, подвешенные в складку на расстоянии 15-20 см от ограждения. Ширина занавеси должна быть в 2 раза больше ширины окна.

2) Отсутствие оконных проемов влечет за собой отсутствие или недостаток естественного освещения. Отсутствие естественного света на рабочем месте влечет за собой снижение общего тонуса организма и ослаблению иммунитета. Более того, отсутствие солнечных лучей негативно влияет на нервную систему и обменные процессы. Для оценки использования естественного света введено понятие коэффициента естественной освещенности (КЕО) и установлены минимальные допустимые значения КЕО — это отношение освещенности E_v внутри помещения за счет естественного света к наружной освещенности E_n от всей полусферы небосклона, выраженное в процентах:

$$KEO = (E_B / E_H) 100\%.$$

По характеристике зрительской работы труд учащихся можно отнести ко второму разряду работы, и при боковом естественном освещении в аудитории, лаборатории на рабочих столах и партах должен обеспечиваться $KEO = 1,5\%$. К средствам нормализации освещения производственных помещений и рабочих мест относятся: источники света, осветительные приборы, световые проемы, светозащитные устройства, светофильтры.

3) Недостаточное освещение влияет на функционирование зрительного аппарата, то есть определяет зрительную работоспособность, на психику человека, его эмоциональное состояние, вызывает усталость центральной нервной системы, возникающей в результате прилагаемых усилий для опознания четких или сомнительных сигналов. Известно, что при длительной работе в условиях недостаточной освещенности и при нарушении других параметров световой среды зрительное восприятие снижается, развивается близорукость, болезнь глаз, появляются головные боли. Норма освещенности, согласно нормативным документам, должна быть не менее 200-300 Лк. Средства нормализации освещения производственных помещений и рабочих мест: источники света; осветительные приборы; световые проемы; светозащитные устройства; светофильтры.

4) Перегрузки в работе систем, а так же несоблюдение правил по электробезопасности могут вызвать повышенное значение напряжения в электрической цепи. Это может повлечь за собой местные поражения организма человека электрическим током (ожоги, механические повреждения и т.п.) или электрический удар. Чтобы избежать короткого замыкания, а значит, возникновения пожара и получения электротравмы, помещения, где размещаются рабочие места с компьютерами, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации.

8.4.Обоснование мероприятий по снижению уровней воздействия опасных и вредных факторов на исследователя.

Организационные мероприятия при компоновке рабочей зоны Большое значение для профилактики статических физических перегрузок имеет

правильная организация рабочего места человека, работающего с ПЭВМ. Рабочее место должно быть организовано в соответствии с требованиями стандартов, технических условий и (или) методических указаний по безопасности труда. Оно должно удовлетворять следующим требованиям:

- 1) обеспечивать возможность удобного выполнения работ;
- 2) учитывать физическую тяжесть работ;
- 3) учитывать размеры рабочей зоны и необходимость передвижения в ней работающего;
- 4) учитывать технологические особенности процесса выполнения работ.

Невыполнение требований к расположению и компоновке рабочего места может привести к получению работником производственной травмы или развития у него профессионального заболевания. Рабочее место программиста должно соответствовать требованиям СанПин 2.2.2/2.4.1340-03. Конструкция оборудования и рабочего места при выполнении работ в положении сидя должна обеспечивать оптимальное положение работающего, которое достигается регулированием высоты рабочей поверхности, высоты сидения, оборудованием пространства для размещения ног и высотой подставки для ног. Схемы размещения рабочих мест с персональными компьютерами должны учитывать расстояния между рабочими столами с мониторами: расстояние между боковыми поверхностями мониторов не менее 1,2 м, а расстояние между экраном монитора и тыльной частью другого монитора не менее 2,0 м. Клавиатура должна располагаться на поверхности стола на расстоянии 100-300 мм от края, обращенного к пользователю. Быстрое и точное считывание информации обеспечивается при расположении плоскости экрана ниже уровня глаз пользователя, предпочтительно перпендикулярно к нормальной линии взгляда (нормальная линия взгляда 15 градусов вниз от горизонтали). Рабочие места с ПЭВМ при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, рекомендуется изолировать друг от друга перегородками высотой 1,5 - 2,0 м.

Разрабатываемый в ходе выполнения ВКР программный продукт используется для анализа численных данных и наборов значений. При работе с данным продуктом не требуется постоянное его использование, так как программа анализирует входные данные и получает необходимый результат, который, непосредственно, будет использован в дальнейшем. Работнику необходимо контролировать параметры входных данных и фиксировать выходные параметры. Следовательно, преимуществом данного продукта является практически полная автономность.

8.5. Экологическая безопасность

Охрана окружающей среды – это комплексная проблема и наиболее активная форма её решения – это сокращение вредных выбросов промышленных предприятий через полный переход к безотходным или малоотходным технологиям производства.

С точки зрения потребления ресурсов компьютер потребляет сравнительно небольшое количество электроэнергии, что положительным образом сказывается на общей экономии потребления электроэнергии в целом.

Основными отходами при выполнении данной бакалаврской работы являются черновики бумаги, отработавшие люминесцентные лампы и картриджи. Израсходованная бумага была направлена на утилизацию в пункт приема макулатуры, люминесцентные лампы – в упакованном виде на ртутьперерабатывающий завод, расположенный в г. Северск. Составляющие израсходованных картриджей (стальные винты, алюминиевые каркасы, пластик, углеродистый порошок) по отдельности в упакованном виде также были направлены на утилизацию в соответствии с требованиями ГОСТ 30775-2001.

8.6. Безопасность в чрезвычайных ситуациях

В Томске преобладает континентально-циклонический климат. Природные явления (землетрясения, наводнения, засухи, ураганы и т. д.) отсутствуют. Возможными ЧС могут являться сильные морозы.

Для Сибири в зимнее время года характерна очень низкая температура воздуха. Достижение критически низких температур приведет к авариям систем

теплоснабжения и жизнеобеспечения, приостановке работы, обморожениям и даже жертвам среди населения. В случае переморозки труб должны быть предусмотрены запасные обогреватели. Их количества и мощности должно хватать для того, чтобы работа на производстве не прекратилась. Кроме того, необходимо иметь альтернативные источники тепла, электроэнергии и транспорта.

Чрезвычайные ситуации, возникающие в результате диверсий, возникают все чаще. Зачастую такие угрозы оказываются ложными. Но случаются взрывы и в действительности.

8.7. Выводы и рекомендации

Проанализировав условия труда на рабочем месте, где была разработана бакалаврская работа, можно сделать вывод, что помещение удовлетворяет необходимым нормам и в случае соблюдения техники безопасности и правил пользования компьютером работа в данном помещении не приведет к ухудшению здоровья работника. Само помещение и рабочее место в нем удовлетворяет всем нормативным требованиям. Кроме того, действие вредных и опасных факторов сведено к минимуму, т.е. микроклимат, освещение и электробезопасность соответствуют требованиям, предъявленным в соответствующих нормативных документах.

Относительно рассмотренного вопроса об экологической безопасности можно сказать, что деятельность в ходе выполнения выпускной квалификационной работы не представляет опасности окружающей среде.

ГЛАВА 9.ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ

Целью написания раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» является проектирование и создание конкурентоспособных разработок, отвечающих современным требованиям в области ресурсоэффективности и ресурсосбережения.

9.1 Оценка коммерческого и инновационного потенциала научных исследований

9.1.1 Потенциальные потребители результатов исследований

Выполненная работа направлена на разработку и реализацию алгоритма, который будет решать задачи линейного программирования с использованием оператора-проектора.

Данная работа имеет высокий потенциал. Описанный в работе алгоритм может использоваться для, например, решения транспортных задач, задач о смесях и других задач линейного программирования. Данный способ имеет ряд преимуществ перед обычно применяемым симплекс – методом Дж.Данцига, такие как:

1. За счет анализа ограничений равенств, существенно сокращается размерность пространства решения задачи.
2. В обычно применяемом методе Данцига каждое ограничение - равенство вида $a^T x = b$ преобразуется в два ограничения неравенства ($a^T x \leq b$, $a^T x \geq b$), что приводит к увеличению числа вершин, которое равно числу сочетаний из n по k , где n – число ограничений неравенств, а k - размерность пространства решения задачи. Увеличение числа вершин многогранника допустимых значений, в свою очередь, может привести к увеличению числа шагов перемещения рабочей точки в точку оптимума.
3. Увеличение числа шагов в поиске оптимума, в свою очередь, приводит к увеличению ошибки округления решения задачи.

В таблице приведены основные сегменты рынка по следующим критериям: размер компании-заказчика и направление деятельности. Анализ рынка

проводился на основе следующих компаний: DPD (крупная компания), Томские Транспортные Линии (средняя компания), ПЭК Томск (мелкая компания).

Таблица 2 – Карта сегментирования рынка

		Направление деятельности	
		Потребности	Запасы
Размер компании	Крупная	+	+
	Средняя	+	–
	Мелкая	+	–

Согласно карте сегментирования рынка, выберем следующие сегменты: построение кластеров изображений для крупных компаний.

9.1.2 Анализ конкурентных технических решений

Анализ конкурентных технических решений позволяет провести оценку сравнительной эффективности научной разработки и определить направления для ее будущего повышения. Основным методом, используемым в данной работе, является метод с использованием оператора - проектора, который используется в поиске оптимума при попадании в многогранник допустимых значений(осн). Конкурентные методы: симплекс – метод (см), простой перебор (пп).

Оценочная карта для сравнения конкурентных технических решений представлена в таблице 3.

Таблица 3 – Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес	Баллы			Конкуренто- способность		
		Б _{осн}	Б _{см}	Б _{пп}	Б _{осн}	Б _{см}	Б _{пп}
Технические критерии оценки ресурсоэффективности							
Эффективность	0,3	4	4	4	1,2	1,2	1,2
Устойчивость	0,2	5	5	5	1	1	1
Временные затраты	0,2	5	4	4	1	0,8	0,8
Новизна метода	0,1	5	3	3	0,5	0,3	0,3
Простота реализации	0,1	5	4	5	0,5	0,4	0,5
Универсальность	0,1	4	4	4	0,5	0,4	0,4
Итого	1	29	24	25	4,7	4,1	4,2

По полученным результатам можно сделать вывод, что разрабатываемый алгоритм для оценки информативности является по конкурентоспособности наиболее эффективным.

9.1.3 SWOT – анализ

SWOT-анализ – это эффективный инструмент стратегического менеджмента. SWOT-анализ заключается в исследовании внешней и внутренней сред проекта.

Факторы разделяются на четыре категории:

- Strengths (сильные стороны),
- Weaknesses (слабые стороны),
- Opportunities (возможности),
- Threats (угрозы).

Разработанная для алгоритма поиска информативного атрибута матрица SWOT представлена в таблице 4.

Таблица 4 – SWOT- анализ

	Сильные стороны: С1. Высокая эффективность алгоритма для решения задач ЛП. С2. Гибкость алгоритма с точки зрения считывания исходных данных. С3. Простая эксплуатация.	Слабые стороны: Сл1. Узкий круг целевой аудитории. Сл2. Трудоемкий процесс написания и отладки программы для решения задач ЛП.
Возможности: В1. Расширение функционала. В2. Написание алгоритма на других языках программирования.	Благодаря гибкости и высокой эффективности алгоритм может быть дополнен реализацией других задач, например, задач квадратичного программирования.	Написание алгоритма на другом языке программирования, а также расширение функционала могут сделать алгоритм более широко применимым, увеличить целевую аудиторию.
Угрозы: У1. Отсутствие спроса на продукт на рынке. У2. Развитие и появление аналогов алгоритма.	Наглядные результаты использования алгоритма, а также его низкая стоимость могут увеличить спрос на него.	Узкая направленность и использование платного языка программирования могут ослабить интерес покупателей.

9.2 Определение возможных альтернатив проведения научных исследований

Для выявления возможных альтернатив разработки проекта и доработки результатов был использован морфологический подход. Он основан на подборе возможных решений для отдельных частей задачи и последующем систематизированном получении их сочетаний.

В таблице 5 в виде матрицы представлены возможные варианты реализации разработки.

Таблица 5 – Морфологическая матрица

	1	2	3
А. Количество языков программирования, для которых проводится анализ	1	2	Больше 2
Б. Платформа для анализа кода	C++	Python	–

Путём комбинации различных параметров были определены два наиболее оптимальных варианта исполнения:

- А1Б1;
- А1Б2.

Для данной матрицы наиболее оптимальным вариантом исполнения является А1Б1.

9.3 Планирование научно-исследовательских работ

9.3.1 Структура работ в рамках научного исследования

Трудоемкость выпускной квалификационной работы определяется опытным путем в человеко-днях и имеет вероятностный характер, так как зависит от трудно учитываемых факторов. Для реализации проекта необходимы: студент-дипломник (СД) и научный руководитель (НР). Основные этапы выполнения ВКР приведены в таблице 6.

Таблица 6 – Основные этапы выполнения ВКР

№	Описание этапа	Исполнители	Загруженность исполнителей
1	Составление и утверждение задания	НР	НР – 100%
2	Анализ предметной области	С, НР	С – 50%, НР – 50%
3	Разработка календарного плана	С, НР	С – 10%, НР – 90%
4	Обзор литературы и интернет - источников	С, НР	С – 50%, НР – 50%
5	Обзор задач и методов решения задач ЛП	С	С – 100%
6	Математическая постановка задачи	С, НР	С – 10%, НР – 90%
7	Выбор метода решения поставленной задачи	С, НР	С – 10%, НР – 90%
8	Разработка алгоритмов решения задачи	С	С – 100%
9	Программная реализация компьютерной модели	С	С – 100%
10	Проведение исследований на модельных данных	С, НР	С – 80% НР – 20%
11	Проведение исследований на реальных данных	С, НР	С – 60% НР – 40%
12	Расчет экономической эффективности научно-технической продукции	С	С – 100%
13	Оценка социальной ответственности проекта	С	С – 100%
14	Написание пояснительной записки	С, НР	С – 90% НР – 10%

9.3.2 Определение трудоемкости выполнения работ

Трудовые затраты в большинстве случаев образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоемкости работ каждого из участников научного исследования.

Ожидаемая продолжительность этапа рассчитывается по формуле:

$$t_{ож} = \frac{3 \cdot t_{\min} + 2 \cdot t_{\max}}{5}, \quad (15)$$

где t_{\min} – предположительно минимальная продолжительность этапа в рабочих днях, определяемая методом экспертной оценки;

t_{\max} – предположительно максимальная продолжительность этапа в рабочих днях, определяемая методом экспертной оценки.

Продолжительность каждого этапа рассчитывается по формуле:

$$t_{\text{раб}} = t_{\text{ож}} \cdot k_{\text{д}}, \quad (16)$$

где $t_{\text{раб}}$ – длительность этапов в рабочих днях;

$k_{\text{д}}$ – коэффициент, учитывающий дополнительное время на консультации и согласование работ, $k_{\text{д}} = 1.2$.

9.3.3 Разработка графика проведения научного исследования

В данном пункте необходимо построить диаграмму Ганта – горизонтально-ленточный график, на котором работы представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения приведенных работ.

Линейный график строится на основании полученных значений $t_{\text{раб}}$, предварительно переведенных в календарные дни по формуле:

$$t_{\text{к}} = t_{\text{раб}} \cdot K_{\text{н}}, \quad (17)$$

где $t_{\text{к}}$ – длительность этапов работ в календарных днях;

$K_{\text{н}}$ – коэффициент календарности.

Коэффициент календарности рассчитывается по формуле:

$$K_{\text{н}} = \frac{T_{\text{к}}}{T_{\text{к}} - T_{\text{вд}} - T_{\text{пд}}}, \quad (18)$$

где $T_{\text{к}}$ – количество календарных дней, $T_{\text{к}} = 365$,

$T_{\text{вд}}$ – количество выходных дней, $T_{\text{вд}} = 108$,

$T_{\text{пд}}$ – количество праздничных дней, $T_{\text{пд}} = 10$,

$$K_{\text{н}} = \frac{365}{365 - 108 - 10} = 1.478. \quad (19)$$

Все расчеты сведены в таблицу 7.

Таблица 7 – Временные показатели проведенных работ

№	Исполнители	Продолжительность работ в днях			Трудоемкость	
		t_{min}	t_{max}	$t_{ож}$	$t_{раб}$	t_k
1	НР	1	2	1,4	1,68	2,48
2	С, НР	2	3	2,4	2,88	4,26
3	С, НР	1	2	1,4	1,68	2,48
4	С, НР	3	5	3,8	4,56	6,74
5	С	2	5	3,2	3,84	5,68
6	С, НР	3	6	4,2	5,04	7,45
7	С, НР	5	8	6,2	7,44	11,00
8	С	3	6	4,2	5,04	7,45
9	С	4	6	4,8	5,76	8,51
10	С, НР	2	5	3,2	3,84	5,68
11	С, НР	2	4	2,8	3,36	4,97
12	С	5	7	5,8	6,96	10,29
13	С	4	5	4,4	5,28	7,80
14	С, НР	9	12	10,2	12,24	18,09
Итого				58	69,6	102,87

На основании таблицы 7 построим диаграмму Ганта (Рис. 1).



■ – инженер

■ – руководитель

Рисунок 1 Диаграмма Ганта

9.4 Бюджет научно-технического исследования (НТИ)

9.4.1 Расчет затрат на специальное оборудование для научных (экспериментальных) работ

В данную статью включают все затраты, связанные с приобретением специального оборудования, необходимого для проведения работ по конкретной теме.

В ходе выполнения НТИ использовалось имеющееся компьютерное оборудование, поэтому его стоимость учитывается в калькуляции в виде амортизационных отчислений за 5 месяцев (таблица 8).

Таблица 8 – Расчет бюджета затрат на приобретение спецоборудования для научных работ

Наименование	Единица измерения	Количество	Цена за единицу, руб.	Затраты на материалы, руб.
Амортизация оборудования	шт.	1	20000	20000

9.4.2 Основная заработная плата исполнителей темы

Основная заработная плата руководителя рассчитывается на основании отраслевой оплаты труда (оклад, стимулирующие выплаты, районный коэффициент). Для студента-дипломника основную заработную плату составляет государственная стипендия с учетом районного коэффициента.

Для расчета основной заработной платы (табл. 9) необходимо привести действительный годовой фонд рабочего времени руководителя и студента (табл. 10).

Таблица 9 – Баланс времени

Показатели рабочего времени	Руководитель	Студент-дипломник
Календарное число дней	365	365
Количество нерабочих дней (выходные дни и праздничные дни)	52 и 10	108 и 10
Потери рабочего времени (отпуск, больничные), дни	48	24
Действительный годовой фонд рабочего времени, дни	255	223

Таблица 10 – Расчет основной заработной платы

Исполнитель	Тарифная заработная плата, руб.	Районный коэффициент	Месячный должностной оклад работника, руб.	Среднедневная заработная плата, руб.	Продолжительность работ, дни	Заработная плата основная, руб.
Руководитель	23265	1,3	30244,5	1218	43	52374
Итого:						52374

9.4.3 Дополнительная заработная плата

Затраты по дополнительной заработной плате исполнителей темы учитывают величину предусмотренных Трудовым кодексом РФ доплат за отклонение от нормальных условий труда, а также выплат, связанных с обеспечением гарантий и компенсаций (при исполнении государственных и общественных обязанностей, при совмещении работы с обучением, при предоставлении ежегодного оплачиваемого отпуска и т.д.).

Расчет дополнительной заработной платы ведется по следующей формуле:

$$Z_{доп} = k_{доп} \cdot Z_{осн} = 0.15 \cdot 52374 = 7856.1, \quad (20)$$

где $Z_{осн}$ – основная заработная плата, $k_{доп}$ – коэффициент дополнительной заработной платы на стадии проектирования, принимается равным 0,15.

9.4.4 Отчисления во внебюджетные фонды (страховые отчисления)

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органами государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$Z_{внеб} = k_{внеб} \cdot (Z_{осн} + Z_{доп}), \quad (21)$$

где $k_{\text{внеб}}$ – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

Для учреждений, осуществляющих образовательную и научную деятельность в 2019г., водится пониженная ставка 28% (п. 6 ч. 1 ст. 58 Закона 212-ФЗ).

Расчет отчислений во внебюджетные фонды приведен в таблице 11.

Таблица 11 – Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.	Дополнительная заработная плата, руб.
Руководитель проекта	52374	7856,1
Коэффициент отчисления во внебюджетные фонды	14664,72	2199,7
Итого		16864,42

9.4.5 Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергии и т.д.

Их величина определяется по следующей формуле:

$$Z_{\text{накл}} = (\text{сумма статей } 1 \div 4) \cdot k_{\text{ир}}, \quad (22)$$

где $k_{\text{ир}}$ – коэффициент, учитывающий накладные расходы, примем равным 16%.

9.4.6 Формирование бюджета затрат научно-исследовательского проекта

Рассчитанная величина затрат научно-исследовательской работы является основой для формирования бюджета затрат проекта, который при формировании договора с заказчиком защищается научной организацией в качестве нижнего предела затрат на разработку научно-технической продукции.

Определение бюджета затрат на научно-исследовательский проект приведено в таблице 12.

Таблица 12 – Расчет бюджета затрат

Наименование статьи	Сумма, руб.
1. Затраты на специальное оборудование	16 000
2. Затраты по основной заработной плате исполнителей проекта	52 374
3. Затраты по дополнительной заработной плате исполнителей проекта	7856
4. Отчисления во внебюджетные фонды	16864
5. Накладные расходы	14 895
Бюджет затрат НТИ	107 989

9.5 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Интегральный финансовый показатель разработки определяется как:

$$I_{финр}^{испi} = \frac{\Phi_{pi}}{\Phi_{max}}, \quad (23)$$

где Φ_{pi} – стоимость варианта исполнения, Φ_{max} – максимальная стоимость исполнения научно-исследовательского проекта.

Т.к. стоимость всех вариантов исполнения одинакова, интегральные финансовые показатели также будут одинаковы и равны 1.

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{pi} = \sum a_i \cdot b_i, \quad (24)$$

где a_i – весовой коэффициент варианта исполнения разработки, b_i – балльная оценка варианта исполнения разработки.

Интегральный показатель эффективности вариантов исполнения разработки определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формулам:

$$I_{исп.1} = \frac{I_{p-исп.1}}{I_{финр.1}},$$

$$I_{исп.2} = \frac{I_{p-исп.2}}{I_{финр.2}}.$$
(25)

Так как интегральные финансовые показатели одинаковы и равны 1, то интегральные показатели эффективности вариантов исполнения разработки равны соответствующим интегральным показателям ресурсоэффективности.

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта и выбрать наиболее целесообразный вариант из предложенных.

Сравнительная эффективность проекта:

$$\mathfrak{E}_{cp} = \frac{I_{исп.i}}{I_{исп.1}}.$$
(26)

В пункте 4.2 было рассмотрено два варианта исполнения алгоритма. На основании этого необходимо провести сравнительную характеристику вариантов исполнения (табл. 13).

Таблица 13– Сравнительная оценка характеристик вариантов исполнения проекта

Критерии	Весовой коэффициент	Исп. 1	Исп. 2
1. Повышение производительности труда	0,3	5	5
2. Удобство в эксплуатации	0,2	4	3
3. Удобство в считывании исходных данных	0,2	5	2
4. Скорость работы	0,1	4	3
5. Простота эксплуатации	0,1	4	3
6. Техническая поддержка платформы	0,1	5	3
I_{pi}		4,6	3,4

На основании полученных показателей выполним сравнение интегрального показателя эффективности вариантов исполнения разработки (табл. 14).

Таблица 14 – Сравнительная эффективность разработки

Показатели	Исп.1	Исп.2
Интегральный финансовый показатель разработки	1	1
Интегральный показатель ресурсоэффективности разработки	4,6	3,4
Интегральный показатель эффективности	4,6	3,4
Сравнительная эффективность вариантов исполнения	1	0,74

С позиции финансовой и ресурсной эффективности на основании таблицы, первый вариант исполнения системы наиболее выгодный. Данный вариант исполнения и используется в выпускной квалификационной работе.

10. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лекция 1. Примеры задач линейного программирования. [Электронный ресурс]. URL: <https://studfiles.net/preview/1722609/> (дата обращения 14.05.2019).
2. Области применения и ограничения использования линейного программирования для решения экономических задач – Линейное программирование в экономике [Электронный ресурс]. – URL: https://otherreferats.allbest.ru/economy/006_5264_0.html (дата обращения 17.05.2019).
3. Дж. Данциг. Линейное программирование, его обобщения и применения. М.: «Прогресс», 1966. с. 96 - 115.
4. В.Н. Задорожный, В.Ф. Зальмеж, А.Ю. Трифонов, А.В. Шаповалов. Высшая математика для технических университетов. Часть I. Линейная алгебра. Т.: «Томский политехнический университет», 2014. с. 76 – 82.
5. Г. Стренг. Линейная алгебра и ее применения. М.: «Мир», 1980 с. 353 – 407.
6. Вылегжанин О.Н., Шкатова Г.И. Учет ограничений равенств при решении оптимизационных задач с линейными ограничениями // Известия Томского политехнического университета. – 2008. с. 76 – 78
7. Вылегжанин О.Н., Шкатова Г.И. Сравнительная оценка двух методов выбора наилучших линейных регрессоров 1988. – С. 18–22.
8. Вылегжанин О.Н., Шкатова Г.И. Решение задачи линейного программирования с использованием оператора-проектора. // Известия Томского политехнического университета. – 2009. с. 37 – 40
9. Гантмахер Ф.Р. Теория матриц. – 5-е изд. – М.: Физматлит, 2004. – 559 с.
10. Гилл Ф., Мюррей У., Райт М. Практическая оптимизация — М.: «Мир», 1985. — 509 с., ил.
11. СН 2.2.4/2.1.8.562–96. Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории застройки.

12. СанПиН 2.2.548-96. Гигиенические требования к микроклимату производственных помещений.
13. СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы.
14. Гигиенические требования к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работы. Выписки из СанПиН 2.2.2.542-96.
15. СНиП 23-05-95 Естественное и искусственное освещение.
16. ГОСТ 12.1.019 (с изм. №1) ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты.
17. Федеральный закон Российской Федерации от 22 июля 2008 г. N 123-ФЗ "Технический регламент о требованиях пожарной безопасности".
18. Федеральный классификационный каталог отходов [Электронный ресурс] – 2013. – Режим доступа: <http://www.ecoguild.ru/>, свободный. – Загл. с экрана.
19. Об утверждении правил обращения с отходами производства и потребления в части осветительных устройств, электрических ламп, ненадлежащие сбор, накопление, использование, обезвреживание, транспортирование и размещение которых может повлечь причинение вреда жизни, здоровью граждан, вреда животным, растениям и окружающей среде: Постановление Правительства Российской Федерации от 3 сентября 2010 года № 681.
20. ТОИ Р-45-084-01. Типовая инструкция по охране труда при работе на персональном компьютере.

ПРИЛОЖЕНИЯ

Приложение А – Процедура перемножения матриц

```
void m_mult (double **A, int ma, int na, double **B, int mb, int nb, double **F)
{
    for (int i = 0; i < ma; i++) {
        for (int j = 0; j < nb; j++) {
            F[i][j] = 0;
            for (int j1 = 0; j1 < mb; j1++) {
                F[i][j] = F[i][j] + (A[i][j1] * B[j1][j]);
            }
        }
    }
}
```

Приложение Б – Вывод матриц на экран

```
void m_print (double **A, int m, int n, string txt) {  
    cout<<endl<<endl<<txt;  
    for(int i=0; i<m; ++i){  
        for(int j=0; j<n; ++j){  
            if (j==0) cout<<endl;  
            cout<<A[i][j]<<" "; }  
    }
```

Приложение В – Поиск столбца с максимальной нормой

```
void max_norm (double **A, int m, int n, int &k, double &m_norm){

    double **a = new double*[m];
    for(int i=0; i<m; ++i){
        a[i] = new double[1];}
    double **aT = new double*[1];
    for(int i=0; i<1; ++i){
        aT[i] = new double[m];}
    double **F = new double*[1];
    for(int i=0; i<1; ++i){
        F[i] = new double[1];}

    double c_norm = 0; //норма текущего столбца (current norm)
    m_norm = 0; //максимальная норма для данной итерации (maximum norm)
    cout<<endl;
    for(int j=0; j<n; ++j){
        c_norm = 0;
        for(int i=0; i<m; ++i){
            a[i][0] = A[i][j];
            aT[0][i] = a[i][0];}
        m_mult(aT, 1, m, a, m, 1, F);
        c_norm = F[0][0];
        if (m_norm < (sqrt(c_norm))) {
            m_norm = sqrt(c_norm);
            k = j;}
        cout<<endl<<"norm of "<<j<<" column = "<<sqrt(c_norm)<<endl;
        cout<<"maximum norm = "<<m_norm<<endl;}
}
```

Приложение Г – Копирование и удаление столбца

```
void col_copy (double **A, double **An, int m, int k, int z) { //копирование столбца
    for(int i=0; i<m; ++i){
        An[i][z-1]=A[i][k];}
}

void col_delete (double **A, int m, int n, int k) { //удаление столбца
    for(int j=k; j<n; ++j){
        for(int i=0; i<m; ++i){
            if (j == n-1) {
                A[i][j] = 0;}
            else
                A[i][j] = A[i][j+1];}}
}
```

Приложение Д – Псевдоинверсия для столбца

```
void col_pseudoinverse (double **A, double **APlus, int m, int z) {  
  
    double **Astr = new double*[m];  
    for(int i=0; i<m; ++i){  
        Astr[i] = new double[1];  
    }  
    double **AstrT = new double*[1];  
    for(int i=0; i<1; ++i){  
        AstrT[i] = new double[m];  
    }  
    double **AstrTd = new double*[1];  
    for(int i=0; i<1; ++i){  
        AstrTd[i] = new double[1];  
    }  
  
    for(int i=0; i<m; ++i){  
        Astr[i][0] = A[i][z];  
        AstrT[0][i] = A[i][z];  
    }  
  
    m_mult(AstrT, 1, m, Astr, m, 1, AstrTd);  
  
    for(int j=0; j<m; ++j){  
        APlus[0][j]=AstrT[0][j]*(1/AstrTd[0][0]);  
    }  
}
```

Приложение Е – Расчёт псевдообратной матрицы

```
void pseudoinversion (double **A, int m, int n, double **APlus) {

    int c=0;
    for(int i=0; i<n; ++i){
        Winv[i] = new double[n];}

    MatDoub A2 (m, n);

    for(int i=0; i<m; ++i){
        for(int j=0; j<n; ++j){
            A2[i][j]=A[i][j];} }

    SVD_Alt MakeSVD (A2);

    for(int i=0; i<m; ++i){
        for(int j=0; j<n; ++j){
            U[i][j]=MakeSVD.u[i][j];} }

    for(int i=0; i<n; ++i){
        for(int j=0; j<n; ++j){
            if (i==j){
                if (abs(MakeSVD.w[i])>0.0001){
                    c+=1;}
                W[i][j]= MakeSVD.w[i];}
            else {
                W[i][j]=0;}} }

    for(int i=0; i<n; ++i){
        for(int j=0; j<n; ++j){
            V[j][i]=MakeSVD.v[i][j];} }

    inversion (W, Winv, c);

    for(int i=0; i<c; ++i){
        for(int j=0; j<c; ++j){
            W[i][j]=Winv[i][j];} }

    m_mult(W, n, n, V, n, n, WV);
    m_mult(U, m, n, WV, n, n, UWV);

    m_mult(W, n, n, V, n, n, WV);
    m_mult(U, m, n, WV, n, n, UWV);

    for(int i=0; i<m; ++i){
        for(int j=0; j<n; ++j){
            APlus[j][i]=UWV[i][j];} }
}
```


Приложение Ж – Расчёт проекции начальной точки на ОДЗ

```
void X1_Calc (double **X, int n, double **B, double **A, int m, int z) {  
    double up, all;  
  
    for(int i=0; i<1; ++i){  
        aaT[i] = new double[1];  
  
        for(int i=0; i<m; ++i){  
            for(int j=0; j<n-z; ++j){  
                a[0][j]=A[i][j];  
                aT[j][0]=A[i][j];  
  
                m_mult(a, 1, n-z, X, n-z, 1, aTX);  
                up = B[i][0]-aTX[0][0];  
  
                m_mult(a, 1, n-z, aT, n-z, 1, aaT);  
                all=up/aaT[0][0];  
  
                for(int i=0; i<n-z; ++i){  
                    aTall[i][0]=all*aT[i][0];  
  
                    for(int i=0; i<n-z; ++i){  
                        X[i][0]=X[i][0]+aTall[i][0];  
                    }  
  
                    cout<<endl<<endl<<"-----X1-----";  
                    m_print(X,n-z,1, "X1");  
                }  
            }  
        }  
    }
```

Приложение 3 – Поиск базисного решения

```

void X_basix (double **A, double **X, double **B, int m, int n, int z, int &iterations_opt)
{
    double up, all; bool y=false; int k = 0;

    double **AC = new double*[m-(n-z)];
    for(int i=0; i<m-(n-z); ++i){
        AC[i] = new double[n-z];}
    double **a = new double*[1];
    for(int i=0; i<1; ++i){
        a[i] = new double[n-z];}
    double **Ra = new double*[n-z];
    for(int i=0; i<n-z; ++i){
        Ra[i] = new double[n-z];}
    double **ACPlus = new double*[n-z];
    for(int i=0; i<n-z; ++i){
        ACPlus[i] = new double[m-(n-z)];}
    double **Y = new double*[m];
    for(int i=0; i<m; ++i){
        Y[i] = new double[1];}
    double **aTRa = new double*[n-z];
    for(int i=0; i<n-z; ++i){
        aTRa[i] = new double[1];}
    double **aT = new double*[n-z];
    for(int i=0; i<n-z; ++i){
        aT[i] = new double[1];}
    double **aTX = new double*[1];
    for(int i=0; i<1; ++i){
        aTX[i] = new double[1];}
    double **aRa = new double*[1];
    for(int i=0; i<1; ++i){
        aRa[i] = new double[1];}
    double **AaT = new double*[m];
    for(int i=0; i<m; ++i){
        AaT[i] = new double[1];}
    double **APlusAaT = new double*[n-z];

    for(int i=0; i<n-z; ++i){
        APlusAaT[i] = new double[1];}

    do {
        iterations_opt= iterations_opt+1;
        m_mult(A, m, n-z, X, n-z, 1, Y); //A*X=Y
        k=0;
        for(int i=0; i<m; ++i){
            if(abs(Y[i][0]-(B[i][0]))<=0.001){
                for(int j=0; j<n-z; ++j){
                    AC[k][j]=A[i][j];} //AC - матрица базисных строк
                k+=1;}}
    }
}

```

```

for(int i=0; i<m; ++i){
    for(int j=0; j<n-z; ++j){
        a[0][j]=A[i][j];
        aT[j][0]=A[i][j];}

pseudoinversion (AC, k, n-z, ACPlus);

m_mult(AC, k, n-z, aT, n-z, 1, AaT);
m_mult(ACPlus, n-z, k, AaT, k, 1, APlusAaT);

for(int i=0; i<n-z; ++i){
    Ra[i][0] = aT[i][0] - APlusAaT[i][0];}

m_mult(a, 1, n-z, X, n-z, 1, aTX);
up = B[i][0]-aTX[0][0];
m_mult(a, 1, n-z, Ra, n-z, 1, aRa);
all=up/aRa[0][0];
for(int i=0; i<n-z; ++i){
    aTRa[i][0]=all*Ra[i][0];}
for(int i=0; i<n-z; ++i){
    X[i][0]=X[i][0]+aTRa[i][0];}

} while (k<(n-z-1));

cout<<endl<<endl<<"-----Xb-----";
}

```

Приложение И - Поиск точки оптимума

```
void X_optimal (double **A, int m, int n, int z, double **P, double **X, double **B, int
&iterations_opt) {
    bool check_U = false; int k = 0; double max_U = 10000; int q=0; double max_AU =
-1; int Aq=0; int f=0;

    double **AU = new double*[m];
    for(int i=0; i<m; ++i){
        AU[i] = new double[1];}

    double **AbPlus = new double*[n-z];
    for(int i=0; i<n-z; ++i){
        AbPlus[i] = new double[m];}

    double **Y = new double*[m];
    for(int i=0; i<m; ++i){
        Y[i] = new double[1];}

    double **Ab = new double*[m];
    for(int i=0; i<m; ++i){
        Ab[i] = new double[n-z];}

    double **U = new double*[m];
    for(int i=0; i<m; ++i){
        U[i] = new double[1];}

    double **AbPlusT = new double*[m];
    for(int i=0; i<m; ++i){
        AbPlusT[i] = new double[n-z];}

    double **Alpha = new double*[1];
    for(int i=0; i<1; ++i){
        Alpha[i] = new double[1];}

    double **S = new double*[n-z];
    for(int i=0; i<n-z; ++i){
        S[i] = new double[1];}

    double **APlusA = new double*[n-z];
    for(int i=0; i<n-z; ++i){
        APlusA[i] = new double[1];}

    double **R = new double*[n-z];
    for(int i=0; i<n-z; ++i){
        R[i] = new double[n-z];}

    double **abT = new double*[1];
    for(int i=0; i<1; ++i){
        abT[i] = new double[n-z];}

    double **abTX = new double*[1];
    for(int i=0; i<1; ++i){
        abTX[i] = new double[1];}

    double **abTS = new double*[1];
    for(int i=0; i<1; ++i){
        abTS[i] = new double[1];}

    double **AUPlus = new double*[n-z];
```

```

        for(int i=0; i<n-z; ++i){
            AUPlus[i] = new double[m];}
double **AUPlusA = new double*[n-z];
        for(int i=0; i<n-z; ++i){
            AUPlusA[i] = new double[n-z];}
double **AUPlusAaT = new double*[n-z];
        for(int i=0; i<n-z; ++i){
            AUPlusAaT[i] = new double[1];}
double **RAaT = new double*[n-z];
        for(int i=0; i<n-z; ++i){
            RAaT[i] = new double[1];}
double **aRAaT = new double*[n-z];
        for(int i=0; i<n-z; ++i){
            aRAaT[i] = new double[1];}
double **a = new double*[1];
        for(int i=0; i<1; ++i){
            a[i] = new double[n-z];}
double **aT = new double*[n-z];
        for(int i=0; i<n-z; ++i){
            aT[i] = new double[1];}
double **aX = new double*[1];
        for(int i=0; i<1; ++i){
            aX[i] = new double[1];}

```

```

m_mult(A, m, n-z, X, n-z, 1, Y);

```

```

m_print(A, m-k-f, n-z, "A");

```

```

for(int i=0; i<m; ++i){
    if(abs(Y[i][0]-(B[i][0]))<=0.1){
        for(int j=0; j<n-z; ++j){
            Ab[k][j]=A[i][j];
        for(int j=0; j<n-z; ++j){
            if (i == m-1) {
                A[i][j] = 0;}
            else
                A[i][j] = A[i+1][j];}}
        k+=1;}}

```

```

pseudoinversion (Ab, k, n-z, AbPlus);

```

```

for(int i=0; i<n-z; ++i){
    for(int j=0; j<k; ++j){
        AbPlusT[j][i]=AbPlus[i][j];}}

```

```

m_mult (AbPlusT, k, n-z, X, n-z, 1, U);

```

```

do{

```

```

    iterations_opt=iterations_opt + 1;

```

```

check_U=true;

m_mult(AbPlus, n-z, k, Ab, k, n-z, R);

m_mult(R, n-z, n-z, P, n-z, 1, S);

m_mult(abT, 1, n-z, X, n-z, 1, abTX);
m_mult(abT, 1, n-z, S, n-z, 1, abTS);
Alpha[0][0]=(-abTX[0][0]/abTS[0][0]);
for(int i=0; i<n-z; ++i){
    X[i][0]=X[i][0]+(Alpha[0][0]*S[i][0]);}

m_mult(AbPlusT, k, n-z, R, n-z, 1, U);

m_print(U, k, 1, "U");

for(int i=0; i<k; ++i){
    if(U[i][0]>0.01){
        check_U=false;}}

for(int i=0; i<k; ++i){
    if(U[i][0]>max_U){
        max_U = U[i][0];
        q = i;}}

for(int i=0; i<m-k-f; ++i){
    for(int j=0; j<n-z; ++j){
        a[0][j]=A[i][j];
        aT[j][0]=A[i][j];}}

pseudoinversion (A, m-k-f, n-z, AUPlus);

m_print(AUPlus, n-z, m-k-f, "AUPlus");

m_mult (AUPlus, n-z, m-k-f, A, m-k-f, n-z, AUPlusA);
m_print(AUPlusA, n-z, n-z, "AUPlusA");
m_mult (AUPlusA, n-z, n-z, aT, n-z, 1, AUPlusAaT);
m_print(aT, n-z, 1, "aT");
m_print(AUPlusAaT, n-z, 1, "AUPlusAaT"); */

pseudoinversion (Ab, k-f, n-z, AUPlus); //псевдообратная от базисного
набора

m_mult (AUPlus, n-z, k-f, Ab, k-f, n-z, AUPlusA);
m_print(AUPlusA, n-z, n-z, "AUPlusA");
m_mult (AUPlusA, n-z, n-z, aT, n-z, 1, AUPlusAaT);
m_print(aT, n-z, 1, "aT");
m_print(AUPlusAaT, n-z, 1, "AUPlusAaT");

for(int i=0; i<n-z; ++i){
    RAaT[i][0] = aT[i][0]-AUPlusAaT[i][0];} //Ra
m_print(RAaT, n-z, 1, "RAaT");

```

```

for(int i=0; i<m-k-f; ++i){
    for(int j=0; j<n-z; ++j){
        a[0][j]=A[i][j];
        aT[j][0]=A[i][j];}

    m_mult(a, 1, n-z, X, n-z, 1, aX);
    m_print(aX, 1, 1, "aX");
    m_mult(a, 1, n-z, RAaT, n-z, 1, aRAaT);
    m_print(aRAaT, 1, 1, "aRAaT");
    AU[i][0] = (B[i][0]-aX[0][0])/(aRAaT[0][0]); //коэффициенты Ла-
гранжа для матрицы A2
    if ((AU[i][0]<max_AU) && (AU[i][0]>=0)){ //поиск строки с ми-
нимальным положительным коэффициентом Лагранжа
        max_AU = AU[i][0];
        Aq = i;}}

m_print(AU, m-k-f, 1, "AU");

for(int i=0; i<n-z; ++i){
    Ab[q][n] = A[Aq][i];}
for(int i=0; i<m-k-f; ++i){
    for(int j=0; j<n-z; ++j){
        if (i == m-k-f-1) {
            A[i][j] = 0;}
        else
            A[i][j] = A[i+1][j];}}
f+=1;

m_print(Ab, k, n-z, "Ab");
m_print(A, m-k-f, n-z, "A");

pseudoinversion (Ab, k, n-z, AbPlus);

for(int i=0; i<n-z; ++i){
    for(int j=0; j<k; ++j){
        AbPlusT[j][i]=AbPlus[i][j];}}

if (f==k) //f - количество вырезанных столбцов, k - количество векторов
Ab
    break;
} while (check_U == false);
}

```

Приложение К – Тело программы

```
int _tmain(int argc, _TCHAR* argv[])
{
    int ma1; int ma2; int na; int z = 0; int k = 0; double norm; double tol; //
    Ограничения-равенства A1[ma1, na], Ограничения-неравенства A2[ma2, na], tol - точность,
    norm - максимальная норма
    bool check = true; int iterations_eq = 0; int iterations_opt = 0;

    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    scanf("%lf", &tol);
    scanf("%d", &ma1);
    scanf("%d", &ma2);
    scanf("%d", &na);

    //Выделение памяти
    double **A1 = new double*[ma1];
    for(int i=0; i<ma1; ++i){
        A1[i] = new double[na];}
    double **B1 = new double*[ma1];
    for(int i=0; i<ma1; ++i){
        B1[i] = new double[1];}

    double **A2 = new double*[ma2];
    for(int i=0; i<ma2; ++i){
        A2[i] = new double[na];}
    double **B2 = new double*[ma2];
    for(int i=0; i<ma2; ++i){
        B2[i] = new double[1];}

    double **P = new double*[na];
    for(int i=0; i<na; ++i){
        P[i] = new double[1];}

    double **X = new double*[na];
    for(int i=0; i<na; ++i){
        X[i] = new double[1];}

    double **A1n = new double*[ma1];
    for(int i=0; i<ma1; ++i){
        A1n[i] = new double[na];}
    double **A2n = new double*[ma2];
    for(int i=0; i<ma2; ++i){
        A2n[i] = new double[na];}

    double **A1nPlus = new double*[na];
    for(int i=0; i<na; ++i){
        A1nPlus[i] = new double[ma1];}

    double **C = new double*[ma1];
    for(int i=0; i<ma1; ++i){
```



```

        C[i] = new double[na];}
double **CPlus = new double*[1];
for(int i=0; i<1; ++i){
    CPlus[i] = new double[ma1];}

double **Pn = new double*[na];
for(int i=0; i<na; ++i){
    Pn[i] = new double[1];}

//Заполнение матриц
for(int i=0;i<na;i++){
    for(int j=0;j<1;j++){
        scanf("%f", &P[i][j]);} }

for(int i=0;i<ma1;i++){
    for(int j=0;j<na;j++){
        scanf("%f", &A1[i][j]);} }

for(int i=0;i<ma2;i++){
    for(int j=0;j<na;j++){
        scanf("%f", &A2[i][j]);} }

for(int i=0;i<ma1;i++){
    for(int j=0;j<1;j++){
        scanf("%f", &B1[i][j]);} }

for(int i=0;i<ma2;i++){
    for(int j=0;j<1;j++){
        scanf("%f", &B2[i][j]);} }

for(int i=0;i<na;i++){
    for(int j=0;j<1;j++){
        scanf("%f", &X[i][j]);} }

//Вывод на экран
m_print (A1, ma1, na, "A1");
m_print (A2, ma2, na, "A2");
m_print (B1, ma1, 1, "B1");
m_print (B2, ma2, 1, "B2");
m_print (P, na, 1, "P");
m_print (X, na, 1, "Xo");

max_norm (A1, ma1, na, k, norm); //Поиск максимальной нормы из A1

z=1; //Первая итерация

// Копирование-удаление
col_copy (A1, A1n, ma1, k, z);
col_copy (A2, A2n, ma2, k, z);
col_copy (P, P, 1, k, z);
col_delete (A1, ma1, na, k);
col_delete (A2, ma2, na, k);

```

```

col_delete (P, na, 1, k);

col_pseudoinverse (A1n, A1nPlus, ma1, 0); //псевдообратная к A1n

norm = 1000;

//Рекуррентный процесс
do {
    if (na > z) {

        iterations_eq+=1; //счётчик
        C_calculation (A1, ma1, na, z, A1n, A1nPlus, C); //расчёт C

        m_print (C, ma1, na-z, "C");

        max_norm (C, ma1, na-z, k, norm); //максимальная норма из C

        if ((norm > tol)) {
            z+=1; //Итерация +1

            col_copy (A1, A1n, ma1, k, z);
            col_copy (A2, A2n, ma2, k, z);
            Pn[z-2][0]=P[k][0];
            col_delete (A1, ma1, na, k);
            col_delete (A2, ma2, na, k);
            for(int i=k; i<na; ++i){
                if (i == na-1) {
                    P[i][0] = 0;}
                else
                    P[i][0] = P[i+1][0];}

            col_pseudoinverse (C, CPlus, ma1, k);
            A1nPlus_refind (A1, ma1, na, z, CPlus, A1nPlus, k);
            for(int i=0; i<ma1; ++i){
                A1nPlus[z-1][i]=CPlus[0][i];}

            check = true; //проверка, что матрица A1 не пуста
            for(int i=0; i<ma1; ++i){
                for(int j=0; j<na; ++j){
                    if (A1[i][j] != 0) {
                        check = false;}}}

            if (check == true)
                break;
        } while (norm > tol);
    }

//~

//Выводы на экран
m_print (A1, ma1, na, "A1*"); //Зависимые столбцы A1
m_print (A2, ma2, na, "A2*"); //Зависимые столбцы A2
m_print (A1n, ma1, z, "A1*n"); //Независимые столбцы A1
m_print (A2n, ma2, z, "A2*n"); //Независимые столбцы A2
//~

```

```

//Проверка: пуста ли A1
if (check == true) {

    cout<<endl<<endl<<"Matrix A is empty";
    X1_Calc (X, na, B2, A2n, ma2, 0); //Расчёт проекции в нульмерное про-
странство
    z=0;}

//В противном случае
else {

    P_calculation (A1, ma1, na, z-1, Pn, P, A1nPlus); //Пересчёт матрицы ко-
эффициентов линейного функционала с учётом ограничений-равенств
    m_print (P, na-z, 1, "P*");

    X1_Calc (X, na, B2, A2, ma2, z); //расчёт первой точки x1
    X_basix (A2, X, B2, ma2, na, z, iterations_opt); //расчёт вершины (базис-
ного решения)
    m_print (X, na-z, 1, "Xb");

    X_optimal (A2, ma2, na, z, P, X, B2, iterations_opt); //расчёт усл. экстре-
мума

    double **XT = new double*[1];
    for(int i=0; i<1; ++i){
        XT[i] = new double[na-z];}

    double **PT = new double*[1];
    for(int i=0; i<1; ++i){
        PT[i] = new double[na-z];}
    double **PX = new double*[1];
    for(int i=0; i<1; ++i){
        PX[i] = new double[1];}

    for(int i=0; i<na-z; ++i){
        PT[0][i]=P[i][0];}

    m_mult(PT, 1, na-z, X, na-z, 1, PX);

    cout<<endl<<endl<<"-----ANSWER-----";

    m_print (P, na-z, 1, "P");
    m_print (X, na-z, 1, "Optimal X: ");

    cout<<endl<<endl<<"I(x) = pT*x = "<<PX[0][0]<<endl;
    cout<<endl<<"Quantity of iterations of the constraints-equalities: "<<itera-
tions_eq<<endl;
    cout<<endl<<"Quantity of iterations of the constraints-unequalities: "<<itera-
tions_opt<<endl;}

return 0; }

```